Hoof Information System

Use Case Specification: Camper Signs Up

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 3/6/18 | 1.0 | Initial version | Kyle Casson |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

Hoof Information System 1

Use Case Specification: Camper Signs Up 1

Version <1.0> 1

Revision History 2

Table of Contents 3

Use Case Specification 1: Finding Information 11

1. 1. Finding Information 11

2. 2. Flow of Events 11

3. 3. Special Requirements 11

4. 4. Pre-conditions 11

5. 5. Post-conditions 11

6. 6. Extension Points 11

Use Case Specification 2: Registering an Auction Item 12

7. 1. Registering an Auction Item 12

8. 2. Flow of Events 12

9. 3. Special Requirements 12

10. 4. Pre-conditions 12

11. 5. Post-conditions 13

12. 6. Extension Points 13

Use Case Specification 3: Attending an Event 14

13. 1. Finding Information 14

14. 2. Flow of Events 14

15. 3. Special Requirements 14

16. 4. Pre-conditions 14

17. 5. Post-conditions 15

18. 6. Extension Points 15

Use Case Specification 4: Writing a Grant 16

19. 1. Writing a Grant 16

20. 2. Flow of Events 16

21. 3. Special Requirements 16

22. 4. Pre-conditions 16

23. 5. Post-conditions 16

24. 6. Extension Points 16

Use Case Specification 5: Organizing Events 17

25. 1. Organizing an Event 17

26. 2. Flow of Events 17

27. 3. Special Requirements 17

28. 4. Pre-conditions 17

29. 5. Post-conditions 18

Use Case Specification 6: Deleting an Event 19

30. 1. Deleting an Event 19

31. 2. Flow of Events 19

32. 3. Special Requirements 19

33. 4. Pre-conditions 19

34. 5. Post-conditions 19

35. 6. Extension Points 20

Use Case Specification 7: Modifying an Event 21

36. 1. Finding Information 21

37. 2. Flow of Events 21

38. 3. Special Requirements 21

39. 4. Pre-conditions 21

40. 5. Post-conditions 21

41. 6. Extension Points 22

Use Case Specification 8: Organizational Donation 23

42. 1. Organizational Donation 23

43. 2. Flow of Events 23

44. 3. Special Requirements 23

45. 4. Pre-conditions 23

46. 5. Post-conditions 23

47. 6. Extension Points 24

Use Case Specification 9: <Attend Event> 25

48. 1. Use-Case Name 25

49. 2. Flow of Events 25

50. 3. Pre-conditions 25

51. 4. Post-conditions 25

52. 5. Extension Points 25

Use Case Specification 10: <Log Work> 27

53. 1. Use-Case Name 27

54. 2. Flow of Events 27

55. 3. Pre-conditions 27

56. 4. Post-conditions 27

Use Case Specification 11: Delete Work 28

57. 1. Use-Case Name 28

58. 2. Flow of Events 28

59. 3. Pre-conditions 28

60. 4. Post-conditions 28

Use Case Specification 12: Modify Work 29

61. 29

62. 1. Use-Case Name 29

63. 2. Flow of Events 29

64. 3. Pre-conditions 29

65. 4. Post-conditions 29

Use Case Specification 13: Store User Info 30

66. 1. Use-Case Name 30

67. 2. Flow of Events 30

68. 3. Pre-conditions 30

Use Case Specification 14: Delete User info 31

69. 1. Use-Case Name 31

70. 2. Flow of Events 31

71. 3. Pre-conditions 31

72. 4. Post-conditions 31

Use Case Specification 15: Modify User Info 32

73. 1. Use-Case Name 32

74. 2. Flow of Events 32

75. 3. Pre-conditions 32

76. 4. Post-conditions 32

Use Case Specification 16: System stores and tracks info 33

77. 1. Use-Case Name 33

78. 2. Flow of Events 33

79. 3. Pre-conditions 33

80. 4. Post-conditions 33

Use Case Specification 17: Delete System Stores 34

81. 1. Use-Case Name 34

82. 2. Flow of Events 34

83. 3. Pre-conditions 34

84. 4. Post-conditions 34

Use Case Specification 18: Modify System Stores 35

85. 1. Use-Case Name 35

86. 2. Flow of Events 35

87. 3. Pre-conditions 35

88. 4. Post-conditions 35

Use Case Specification 19: User enters Data 36

89. 1. Use-Case Name 36

90. 2. Flow of Events 36

91. 3. Pre-conditions 36

92. 4. Post-conditions 36

Use Case Specification 20: User Deletes data 37

93. 1. Use-Case Name 37

94. 2. Flow of Events 37

95. 3. Pre-conditions 37

96. 4. Post-conditions 37

Use Case Specification 21: <User Modifies Data> 38

97. 1. Use-Case Name 38

98. 2. Flow of Events 38

99. 3. Pre-conditions 38

100. 4. Post-conditions 38

Use Case Specification 22: Create Account 39

101. 1. Use-Case Name: User Creates Account 39

102. 2. Flow of Events 39

Use Case Specification 23: Donor Makes Donation 40

103. Use-Case Name 40

104. Flow of Events 40

105. Special Requirements 40

106. Pre-conditions 40

107. Post-conditions 40

Use Case Specification 24: Donor Schedules Donation 41

108. Use-Case Name 41

109. Flow of Events 41

110. Special Requirements 41

111. Pre-conditions 41

112. Post-conditions 41

Use Case Specification 25: Donor Cancels Donation 42

113. Use-Case Name 42

114. Flow of Events 42

115. Special Requirements 42

116. Pre-conditions 42

117. Post-conditions 42

Use Case Specification 26: Donor Receives Donation Receipt 43

118. Use-Case Name 43

119. Flow of Events 43

120. Special Requirements 43

121. Pre-conditions 43

122. Post-conditions 43

Use Case Specification 27: Donor Receives Thank You 44

123. Use-Case Name 44

124. Flow of Events 44

125. Special Requirements 44

126. Pre-conditions 44

127. Post-conditions 44

Use Case Specification 28: User Signs up For Newsletter 45

128. Use-Case Name 45

129. Flow of Events 45

130. Special Requirements 45

131. Pre-conditions 45

132. Post-conditions 45

Use Case Specification 29: User Unsubscribes From Newsletter 46

133. Use-Case Name 46

134. Flow of Events 46

135. Special Requirements 46

136. Pre-conditions 46

137. Post-conditions 46

Use Case Specification 30: User Follows HOOF KY on Social Media 47

Use Case Specification 31: Volunteer Modifies Data 48

Use Case Specification 32: Volunteer Deletes Data 49

Use Case Specification 33: Volunteer is Assigned to an Event 50

Use Case Specification 34: Volunteer is Reassigned to Event 51

Use Case Specification 35: Volunteer is Dropped from Event 52

Use Case Specification 36: Volunteer Receives Confirmation 53

Use Case Specification 37: Volunteer Logs Hours 54

Use Case Specification 40: Camper Signs Up 55

138. Use-Case Name: Camper Signs Up 55

139. Flow of Events 55

140. Special Requirements 55

141. Pre-conditions 55

142. Post-conditions 56

Use Case Specification 41: Camper Cancels Camp Event 57

143. Use-Case Name 57

144. Flow of Events 57

145. Special Requirements 57

146. Pre-conditions 57

147. Post-conditions 57

Use Case Specification 42: Board Member Uploads Files 58

148. Use-Case Name 58

149. Flow of Events 58

150. Special Requirements 58

151. Pre-conditions 58

152. Post-conditions 58

Use Case Specification 43: Board Member Modifies Files 59

153. Use-Case Name 59

154. Flow of Events 59

155. Special Requirements 59

156. Pre-conditions 59

157. Post-conditions 59

Use Case Specification 44: Board Member Deletes Files 60

158. Use-Case Name 60

159. Flow of Events 60

160. Special Requirements 60

161. Pre-conditions 60

162. Post-conditions 60

Use Case Specification 45: Modify Auction Item 61

163. Use-Case Name 61

164. Flow of Events 61

165. Special Requirements 61

166. Pre-conditions 61

167. Post-conditions 61

Use Case Specification 46: Delete Auction Item 62

168. Use-Case Name 62

169. Flow of Events 62

170. Special Requirements 62

171. Pre-conditions 62

172. Post-conditions 62

Use Case Specification 1: Finding Information

# 1. Finding Information

## 1.1 Brief Description

Site visitors as well as volunteers will need to be able to access the site and find relevant information records that will stored inside HOOF’s database.

# 2. Flow of Events

## 2.1 Basic Flow

* The customer will navigate to the about us information section when navigating HOOF’s website.
* This information will include:
* About – Basic information on the HOOF organization.
* Mission Statement –HOOF’s mission statement.
* Calendar- Basic overview of upcoming events.
* Location – Directions to the location of HOOF: address.
* Contact – Contact information: email and phone number
* Board Members – A list of current and active board members.

## 2.2 Alternative Flows

### 2.2.1 Site visitors find outdated information Throughout the site and can cause a misconnect amongst visitors and volunteers

* Admins must enter correct “about” information, so HOOF KY can be referenced properly

# 3. Special Requirements

## · HOOF must have working contact us button with relevant information being pulled from a database or third-party service.

## · HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

# 4. Pre-conditions

* Visitors and Volunteers will not clearance to access basic information on HOOF KY, but to access sensitive data they will need to have a unique USER\_ID that will be able to log-in to the site.

# 5. Post-conditions

* Volunteers and visitors will be able to access the specific information they are looking for such as volunteer data and contact data.
* HOOF will have this information on hand and whilst information gets updated; the changes to the site will be rather immediate with the ability to access their new database.

# 6. Extension Points

## 6.1 Confidentiality

### 6.1.1 Any information collected from users via web forms or email will remain as sensitive data that regular visitors and volunteers will not be able to access.

Use Case Specification 2: Registering an Auction Item

# 1. Registering an Auction Item

## 1.1 Brief Description

This case will describe how a person interested in donating an item to HOOF’s silent auction will register their item with HOOF. This will be done through a basic web form that will allow the potential donor to fill out the relevant information as well as notify them of the event/ where to drop off their item.

# 2. Flow of Events

## 2.1 Basic Flow

* The customer will navigate to the auction section once on HOOF’s website.
* Visitor/Donor will access auction tab
* Visitor/Donor will enter unique donor ID
* Visitor/Donor will enter unique password that coincides with user ID.
* Visitor/Donor will access the web form linked within the HOOF website
* Visitor/Donor will enter Auction item information:
  + Name
  + Donor – can remain anonymous
  + Starting Bid
  + Description
  + Donor Notes
* Visitor/Donor will submit item
* Item will be stored in database
* Message will be displayed – “Success” or “Failure” – based upon if fields are completed or connection to website

## 2.2 Alternative Flows

### 2.2.1 Site visitors find the web form is not able to access the database due to user connection or site connection; which can cause a misconnect amongst visitors, volunteers, and admins.

* Under this circumstance many users will be able to submit the item but it will not be confirmed until a correct connection is established between the server and user.

# 3. Special Requirements

## · HOOF must have working auction item button with relevant information being pulled from a database or third-party service. Thus, allowing for timely adding of auction items.

## · HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

# 4. Pre-conditions

* Visitors and Donors will have clearance to access basic information on HOOF KY’s website when adding an auction item, but to access previous auction data they will need to have a unique USER\_ID that will be able to log-in to the site.
* The database must be well organized and able to be quickly accessed based on information updates and querying for the ability to have efficient access to auction items.

# 5. Post-conditions

* Visitors and Donors will be able to access the specific information they entered for their specific auction item and relevant data for auction dates.
* HOOF will have this information on hand and whilst information gets updated; the changes to the database will be rather immediate in regard to auction items being updated.

# 6. Extension Points

## 6.1 Confidentiality

### 6.1.1 Any information collected from users via web forms or email will remain as sensitive data that regular visitors and volunteers will not be able to access.

Use Case Specification 3: Attending an Event

# 1. Finding Information

## 1.1 Brief Description

This case describes how those who wish to attend auctions or other events for HOOF KY will register for the specific event as well as event information.

# 2. Flow of Events

## 2.1 Basic Flow

* The customer will navigate to the register section once on H OOF’s website.
* Visitor/Donor will access register section tab
* Visitor/Donor will enter unique USER ID
* Visitor/Donor will enter unique password that coincides with USER ID.
* Visitor/Donor will access the web form linked within the HOOF website that allows them to register for events
* Visitor/Donor will enter basic event information as well as billing information:
  + Name
  + Select event drop down
  + Select specific event
  + If event has an associated cost – enter billing information
* Visitor/Donor will submit form
* Record will be stored in database
* Message will be displayed – “Success” or “Failure” – based upon if fields are completed or connection to website

## 2.2 Alternative Flows

### 2.2.1 Site visitors find outdated information or event information is incorrect throughout the site and can cause a misconnect amongst visitors and volunteers

* Admins must enter correct event information, so HOOF KY can be referenced properly
* Users will be able to submit the web form for attendance, but it will not be confirmed until a correct connection is established between the server and user and a confirmation message displayed.

# 3. Special Requirements

## · HOOF must have working register button with relevant information being pulled from a database or third-party service.

## · HOOF must have an applicable database in which it is able to store all its relevant information such as the items listed within the main flow.

# 4. Pre-conditions

* Visitors and Volunteers will have clearance to access basic information on HOOF KY, but to access sensitive data they will need to have a unique USER\_ID that will be able to log-in to the site.
* The database must be well organized and able to be quickly accessed based on information updates and querying for the ability to have efficient access to events registered for as well as billing information and querying.

# 5. Post-conditions

* Volunteers and visitors will be able to access the specific information entered for their specific event they would like to attend and relevant data for auction/event dates.
* HOOF will have this information on hand and whilst information gets updated; the changes to the site will be rather immediate with the ability to access their new database.

# 6. Extension Points

## 6.1 Confidentiality

### 6.1.1 Any information collected from users via web forms or email will remain as sensitive data that regular visitors and volunteers will not be able to access.

Use Case Specification 4: Writing a Grant

# 1. Writing a Grant

## 1.1 Brief Description

Site describes how a HOOF board member will be able to use Hoof’s new system to pull the needed information to write a grant.

# 2. Flow of Events

## 2.1 Basic Flow

* The will navigate to the login section once on HOOF’s website.
* Board of Directors/admin will access login section tab
* Board of Directors/admin will enter unique USER ID
* Board of Directors/admin will enter unique password that coincides with USER ID.
* Board of Directors/admin will access the web form linked within the HOOF website that allows them to access various forms that are required to write grants
* Download will begin once initiated by user
* File can be preloaded with organization information based upon previous grants

## 2.2 Alternative Flows

### 2.2.1 A Board Director or admin could not have the correct credentials or access the wrong forms which could slow down the grant writing process.

# 3. Special Requirements

## · HOOF must have updated files always present within their site to stay compliant with the United States government tax policies for non-profit organization.

## · HOOF must have an applicable database in which it is able to store all of its media files and be able to support multiple downloads from the server.

# 4. Pre-conditions

* Admins/ Board of Directors will have clearance to access basic information on HOOF KY, but to access sensitive data they will need to have a unique USER\_ID that will be able to log-in to the site.
* The database must be well organized and able to be quickly accessed based on information updates and querying.

# 5. Post-conditions

* Volunteers and visitors will be able to access the specific information they are looking for such as volunteer data and contact data.
* HOOF will have this information on hand and whilst information gets updated; the changes to the site will be rather immediate with the ability to access their new database.

# 6. Extension Points

## 6.1 Confidentiality

### 6.1.1 Any information collected from users via web forms or email will remain as sensitive data that regular visitors and volunteers will not be able to access.

Use Case Specification 5: Organizing Events

# 1. Organizing an Event

## 1.1 Brief Description

Describes how the HOOF board will use the system to plan new events and integrate them into the calendar.

# 2. Flow of Events

## 2.1 Basic Flow

* The Board Members will navigate to the events section when navigating HOOF’s website.
* The Board Members will enter unique donor ID
* The Board Members will enter unique password that coincides with user ID.
* The Board Members will access the web form linked within the HOOF website
* The Board Members will enter event information:
  + Event Name
  + Date
  + Time
  + Description
  + Board Member notes
* The Board Members will submit item
* Item will be stored in database
* Message will be displayed – “Success” or “Failure” – based upon if fields are completed or connection to website

## 2.2 Alternative Flows

### 2.2.1 Board Members post outdated information and don’t garbage collect old events or update progress/ success of past events throughout the site and can cause a misconnect amongst visitors and volunteers

* Admins/ Board Members must enter correct event information, so HOOF KY can be referenced properly

# 3. Special Requirements

## · HOOF must have working calendar button with relevant information being pulled from a database or third-party service.

## · HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

# 4. Pre-conditions

* The Board Members will not have clearance to access basic information on HOOF KY, but to access sensitive data they will need to have a unique USER\_ID that will be able to log-in to the site.
* The database must be well organized and able to be quickly accessed based on information updates and querying.

# 5. Post-conditions

* The Board Members will be able to publish events with specific information that is listed within the main flow.
* HOOF will have this information on hand and whilst information gets updated; the changes to the site will be rather immediate with the ability to access their new database and to keep the sites visitors informed on all event updates.

Use Case Specification 6: Deleting an Event

# 1. Deleting an Event

## 1.1 Brief Description

Describes how the HOOF board will use the system to cancel events.

# 2. Flow of Events

## 2.1 Basic Flow

* The Board Members will navigate to the events section when navigating HOOF’s website.
* The Board Members will enter unique donor ID
* The Board Members will enter unique password that coincides with user ID.
* The Board Members will access the delete function for an even
* Then confirm the deletion of the event
* The Board Members will submit item
* Item will be deleted amongst the site and the database
* Message will be displayed – “Success” or “Failure” – based upon if fields are completed or connection to website

## 2.2 Alternative Flows

### 2.2.1 The board members will not be able to delete an event which will cause outdated information throughout the site and can cause a misconnect amongst visitors and volunteers

# 3. Special Requirements

## · HOOF must have working delete button within the elevated access portal for events button with relevant information being pulled from a database or third-party service.

## · HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

# 4. Pre-conditions

* Visitors and Volunteers will not clearance to access basic information on HOOF KY, but to access sensitive data they will need to have a unique USER\_ID that will be able to log-in to the site.
* The database must be well organized and able to be quickly accessed based on information updates and querying.

# 5. Post-conditions

* Volunteers and visitors will be able to access the specific information they are looking for such as volunteer data and contact data.
* HOOF will have this information on hand and whilst information gets updated; the changes to the site will be rather immediate with the ability to access their new database.

# 6. Extension Points

## 6.1 Confidentiality

### 6.1.1 Any information collected from users via web forms or email will remain as sensitive data that regular visitors and volunteers will not be able to access.

Use Case Specification 7: Modifying an Event

# 1. Finding Information

## 1.1 Brief Description

Site visitors as well as volunteers will need to be able to access the site and find relevant information records that will stored inside HOOF’s database.

# 2. Flow of Events

## 2.1 Basic Flow

* The Board Members will navigate to the events section when navigating HOOF’s website.
* The Board Members will enter unique donor ID
* The Board Members will enter unique password that coincides with user ID.
* The Board Members will access the modify function for an even
* Then confirm the modify of the event
* The Board Members will submit item
* Item will be modified amongst the site and the database
* Message will be displayed – “Success” or “Failure” – based upon if fields are completed or connection to website

## 2.2 Alternative Flows

### 2.2.1 Site visitors find outdated information Throughout the site and can cause a misconnect amongst visitors and volunteers

* Admins must enter correct “about” information, so HOOF KY can be referenced properly

# 3. Special Requirements

## · HOOF must have working contact us button with relevant information being pulled from a database or third-party service.

## · HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

# 4. Pre-conditions

* Visitors and Volunteers will not clearance to access basic information on HOOF KY, but to access sensitive data they will need to have a unique USER\_ID that will be able to log-in to the site.
* The database must be well organized and able to be quickly accessed based on information updates and querying.

# 5. Post-conditions

* Volunteers and visitors will be able to access the specific information they are looking for such as volunteer data and contact data.
* HOOF will have this information on hand and whilst information gets updated; the changes to the site will be rather immediate with the ability to access their new database.

# 6. Extension Points

## 6.1 Confidentiality

### 6.1.1 Any information collected from users via web forms or email will remain as sensitive data that regular visitors and volunteers will not be able to access.

Use Case Specification 8: Organizational Donation

# 1. Organizational Donation

## 1.1 Brief Description

Site visitors as well as volunteers will be able to access the donation page and fill in the

# 2. Flow of Events

## 2.1 Basic Flow

* The Visitors will navigate to the events section when navigating HOOF’s website.
* The Visitors will enter unique donor ID
* The Visitors will enter unique password that coincides with user ID.
* The Visitors will access the donation function
* Then they will fill out the desired information and dollar amount
* The visitor will be redirected to paypal
* The visitor will submit payment
* Message will be displayed – “Success” or “Failure” – based upon if fields are completed or connection to website

## 2.2 Alternative Flows

### 2.2.1 Site visitors find outdated information Throughout the site and can cause a misconnect amongst visitors and volunteers

* Admins must enter correct “about” information, so HOOF KY can be referenced properly

# 3. Special Requirements

## · HOOF must have working third party payment service with relevant information being pulled from a database or third-party service.

## · HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

# 4. Pre-conditions

* Visitors and Volunteers will not clearance to access basic information on HOOF KY, but to access sensitive data they will need to have a unique USER\_ID that will be able to log-in to the site.
* The database must be well organized and able to be quickly accessed based on information updates and querying.

# 5. Post-conditions

* Volunteers and visitors will be able to access the specific information they are looking for such as volunteer data and contact data.
* HOOF will have this information on hand and whilst information gets updated; the changes to the site will be rather immediate with the ability to access their new database.

# 6. Extension Points

## 6.1 Confidentiality

### 6.1.1 Any information collected from users via web forms or email will remain as sensitive data that regular visitors and volunteers will not be able to access.

Use Case Specification 9: <Attend Event>

# 1. Use-Case Name

## 1.1 Brief Description

Primary Actor: Charity Event Attendee

Describes how any person attending a HOOF fundraiser event will be added to the system, and the branching paths for donating or not.

# 2. Flow of Events

## 2.1 Basic Flow

A Charity Event Attendee will be able to sign up for a charity event in advance using the HOOF website. When clicking on the Events tab on the website, the user will be able to create an account. If the user already has an account, they will be able to sign in, and mark themselves as attending the event. Upon creating an account, the system will add the attendee information, including the attendee’s name, phone number, email address, and whether they would like to receive further information from HOOF. Once the Attendee marks themselves as attending the event, that information will also be recorded within the system. If an account is created for an attendee, they will be prompted by the system if they would like to donate. If the Attendee selects “Yes” they will redirected to the “Donate Now” page, where they will complete “Donor Makes Donation.”

## 2.2 Alternative Flows

### 2.2.1 < First Alternative Flow >

If a Charity Event Attendee does not register in advance, and shows up to the event in person, they will be asked in person to create an account. They will provide their information, which will be recorded by the system in the same fashion as in the Basic Flow. They will also be asked if they would like to donate.

# 3. Pre-conditions

## 3.1 < Pre-condition One >

The Events page on the website must prompt users to sign in/create an account, as well as allowing them to mark themselves as attending an event, and asking if they would like to donate. This information must be captured and stored into a database.

# 4. Post-conditions

## 4.1 < Post-condition One >

After the use case has finished, the system will contain information on which donors/supporters are attending which events, as well as whether or not they are a donor.

# 5. Extension Points

## 5.1 <Name of Extension Point>

There are two extension points: Event Attendee and Donor Makes Donation. Event Attendee describes how an Attendee registers for an event, which is lightly described here. Donor Makes Donation describes how a donor/supporter can make a donation, which is an extension of asking whether or not they would like to donate in this use case.

Use Case Specification 10: <Log Work>

# 1. Use-Case Name

## 1.1 Brief Description

Primary Actor: HOOF Volunteer

Describes how a HOOF volunteer will be able to log the actions and time they worked at an event or on a project.

# 2. Flow of Events

## 2.1 Basic Flow

A HOOF Volunteer can log their actions and time worked through a form available once logging into their account on the website. By logging in, they can enter different data, including the event name, date, and time that they worked. Once this information is submitted, the system will take the information and enter it into the database.

## 2.2 Alternative Flows

### 2.2.1 < First Alternative Flow >

At certain events, there may be a computer available where users can log their work immediately, once they finish working.

# 3. Pre-conditions

## 3.1 < Pre-condition One >

There must exist a form that can accept user information, and upon submission record the information into the system database.

# 4. Post-conditions

## 4.1 < Post-condition One >

After the use case has finished, the system will contain information on the date of volunteering, the number of hours, and the type of volunteering done.

## 4.2 <Name of Extension Point>

The Delete Work and Modify Work use cases for volunteers are both extensions for this use case. Delete Work and Modify Work both directly manipulate the data that was originally entered into the system by Log Work.

Use Case Specification 11: Delete Work

# 1. Use-Case Name

## 1.1 Brief Description

· Allows the volunteer to delete any volunteer hours they have logged with into the site if there was a mistake or discrepancy in the hours they logged

# 2. Flow of Events

## 2.1 Basic Flow

· Volunteer navigates to their time sheets.

· Volunteer highlights the hours with in the time sheet they wish to delete.

· Once all hours are selected customer hits the delete button.

# 3. Pre-conditions

## 3.1 Volunteer is logged into their profile on the website

## 3.2 Volunteer must have hours logged

# 4. Post-conditions

## 4.1 Volunteer must confirm they wish to delete the hours

## 4.2 Volunteers time sheet will now reflect the deletion

Use Case Specification 12: Modify Work

# 

# 1. Use-Case Name

## 1.1 Brief Description

· Allows the user to modify the hours that they have already logged if there is a mistake, discrepancy or if they have worked more hours since.

# 2. Flow of Events

## 2.1 Basic Flow

· Volunteer navigates to their time sheets.

· Volunteer selects the modify button

· Volunteer corrects/changes hours worked appropriately

· Once finished the volunteer submits modifies hours

## 2.2 Alternative Flows

### 2.2.1 < First Alternative Flow >

· Volunteer enters invalid amount (e.g. negative number)

· Error message pops up for the volunteer

# 3. Pre-conditions

## 3.1 < Pre-condition One >

· Volunteer is logged into their profile on the website

## 3.2 < Pre-condition Two >

· Volunteer must have hours logged

# 4. Post-conditions

## 4.1 < Post-condition One >

· Volunteer must confirm they wish to delete the hours

· Volunteers time sheet will now reflect the deletion

Use Case Specification 13: Store User Info

# 1. Use-Case Name

## 1.1 Brief Description

Will allow donors to store their personal information into the system.

# 2. Flow of Events

## 2.1 Basic Flow

* Donor enter first name.
* Donor enters middle initial.
* Donor enters last name
* Donor enters street address
* Donor enters city
* Donor enters state
* Donor enters zip

## 2.2 Alternative Flows

### 2.2.1 < First Alternative Flow >

* User enters Invalid address
* System displays error message

# 3. Pre-conditions

## 3.1 < Pre-condition One >

* Donor is logged into their profile on the website

## 3.2 < Pre-condition Two >

* Donor must have hours logged

4. Post-conditions

## 4.1 < Post-condition One >

Donor must confirm they want to submit their personal info.

Use Case Specification 14: Delete User info

# 1. Use-Case Name

## 1.1 Brief Description

Allow Donors to delete any personal information they previously saved

# 2. Flow of Events

## 2.1 Basic Flow

* User Logs into profile
* User navigates to personal info
* User clicks deletes personal info button

# 3. Pre-conditions

## 3.1 < Pre-condition One >

User Must be logged into profile

# 4. Post-conditions

## 4.1 < Post-condition One >

User must confirm they want to delete personal information

## 4.2 < Post-condition Two >

User will no longer have a profile and will be logged out

Use Case Specification 15: Modify User Info

# 1. Use-Case Name

## 1.1 Brief Description

Users will be able to delete their personal info from the system.

# 2. Flow of Events

## 2.1 Basic Flow

* User logs into their profile
* Donor clicks edit personal info button
* User changes a personal info field (first name, middle initial, last name or address)
* User hits the submit button

## 2.2 Alternative Flows

### 2.2.1 < First Alternative Flow >

Donor enters invalid address

Donor receives error message from system

# 3. Pre-conditions

## 3.1 Donor is logged into their profile on the website

## 3.2 Volunteer must have personal info entered

# 4. Post-conditions

[A post-condition of a use case is a list of possible states the system can be in immediately after a use case has finished.]

## 4.1 < Post-condition One >

Donor must Confirm they want to modify their personal information

Use Case Specification 16: System stores and tracks info

# 1. Use-Case Name

## 1.1 Brief Description

The user will submit data to the system (e.g. donor’s address, Volunteers hours, Board Members documents) and the system will back up the data.

# 2. Flow of Events

## 2.1 Basic Flow

User Logs in

User enters, modifies or deletes info

User hits submit button

System receives data

System store data in backup server

## 2.2 Alternative Flows

### 2.2.1 < First Alternative Flow >

* User Logs in
* User enters, modifies or deletes info
* User hits submit button
* System receives data
* System fails to backup info
* System attempts back up once more
* If system fails again, error message sent to the user

# 3. Pre-conditions

## 3.1 < Pre-condition One >

The user must be logged in

# 4. Post-conditions

## 4.1 < Post-condition One >

* Data will be saved and ready to be accessed by a user when needed

Use Case Specification 17: Delete System Stores

# 1. Use-Case Name

## 1.1 Brief Description

The system will check files to see if they have been used in recent years. If they have not been used it will add that file into a report which is then sent to the system master as a recommended file to be deleted and will wait a response from the user

# 2. Flow of Events

## 2.1 Basic Flow

* System searches through access logs
* System identifies file that have not been used in 3 years or more
* System Copies file details into a report
* System sends report to the system master
* The system receives response from system master

## 2.2 Alternative Flows

### 2.2.1 < First Alternative Flow >

* System searches through files
* System identifies duplicate or outdated files (e.g. contact information)
* System deletes redundant files

# 3. Pre-conditions

## 3.1 < Pre-condition One >

All deletions must be approved by system Master

System performs the on specified recurring date

# 4. Post-conditions

## 4.1 < Post-condition One >

All files selected by the system master to be deleted will be erased from the system

Use Case Specification 18: Modify System Stores

# 1. Use-Case Name

## 1.1 Brief Description

· The system will occasionally modify files by updating them to reflect changes made by the users (e.g. volunteer adds more time, Board members adds auction items, etc.)

# 2. Flow of Events

## 2.1 Basic Flow

· User modifies a file/ data

* User hits submit button
* System updates the back up with new data added to the file

## 2.2 Alternative Flows

### 2.2.1 < First Alternative Flow >

· User modifies a file/data

* User hits submit button
* System fails at updating the file
* System sends an error message to the user

# 3. Pre-conditions

## 3.1 < Pre-condition One >

· A file must be edited and submitted before the file may be updated

# 4. Post-conditions

## 4.1 < Post-condition One >

· The old file (before the update information) will be overwritten unless user decides to create a new file rather than update a current file

Use Case Specification 19: User enters Data

# 1. Use-Case Name

## 1.1 Brief Description

* Describes how donors will enter their personal information when making donations/

# 2. Flow of Events

## 2.1 Basic Flow

* Donor enters first name
* Donor enters middle initial
* Donor enters last name
* Donor enters street address
* Donor enters city
* Donor enter state
* Donor enters zip
* Donor enters phone number
* Donor enters email
* Donor chooses what they wish to donate
* Donor clicks the submit button

## 2.2 Alternative Flows

### 2.2.1 < First Alternative Flow >

* Donor enters invalid address, city, state, zip, phone number, and/or email
* System returns error message the user.

# 3. Pre-conditions

## 3.1 < Pre-condition One >

* Donor must be logged in.

## 3.2 < Pre-condition Two >

* Donor must be donating an item

# 4. Post-conditions

## 4.1 < Post-condition One >

* Donor will receive a thank you message from the system

## 4.2 < Post-condition two >

* The system will store all donor info

Use Case Specification 20: User Deletes data

# 1. Use-Case Name

## 1.1 Brief Description

Allow Donors to delete any personal information they previously saved

# 2. Flow of Events

## 2.1 Basic Flow

* Donor navigates to personal info
* Donor clicks deletes personal info button
* Donor decides which fields they would like to delete (if not all)
* Donor Hits the delete button

# 3. Pre-conditions

## 3.1 < Pre-condition One >

User Must be logged into profile

# 4. Post-conditions

## 4.1 < Post-condition One >

User must confirm they want to delete personal information

## 4.2 < Post-condition Two >

If all information is deleted user will no longer have a profile and will be logged out

Use Case Specification 21: <User Modifies Data>

# 1. Use-Case Name

## 1.1 Brief Description

Primary Actor: Donor

Describes how Donors will modify their personal information.

Risk Level: High

# 2. Flow of Events

## 2.1 Basic Flow

A Donor can modify their personal information by logging in to their account, and then selecting an option to “Edit Personal Information.” The resulting screen will have text boxes containing all of the customer’s personal information that they previously entered. This information can be modified, and upon hitting submit, the system will modify the information previously held on the Donor. A notification is sent to both a HOOF employee and the Donor.

## 2.2 Alternative Flows

### 2.2.1 < First Alternative Flow >

A user may also contact support if they would like to modify their personal information without logging into the website (over the phone, or through E-mail).

# 3. Pre-conditions

## 3.1 < Pre-condition One >

The system should be able to modify data based on a user request, as well as automatically send an alert both to HOOF and to the user in question.

# 4. Post-conditions

## 4.1 < Post-condition One >

The user’s donor information will be modified to match what they had entered on the form, and an E-mail will be sent.

## 4.2 <Name of Extension Point>

This is an extension of User Enters Data, in that the data that was previously

Use Case Specification 22: Create Account

# 1. Use-Case Name: User Creates Account

## 1.1 Brief Description

This use case describes how a User would create their own account on the HOOF KY website.

# 2. Flow of Events

## 2.1 Basic Flow

* User enters first name into text box
* User enters last name into text box
* User enters middle initial into text box
* User enters email address into text box
* User enters phone number into text box
* User enters a password into text box
* User re-enters password into “Confirm Password” text box
* User clicks “Create Account” button
* A pop up display asks the user to confirm the information entered
* User clicks “Confirmed” button on the pop up
* A pop up display says “Your account has been created.”

## 1.1 Pre-conditions

## 1.2 User is already on hoof ky create account webpage

## 2.1 Post-conditions

## 2.2 User’s account is saved in the HOOF KY database

## 2.3 User can sign into their account through Error! Hyperlink reference not valid. by clicking “Sign In” and entering their email and password

# Use-Case Name

## Brief Description

This use case describes how donor will donate money and the frequency in which they donate.

# Flow of Events

## Basic Flow

* Donor enters their first name
* Donor enters last name
* Donor enters street address
* Donor enters city
* Donor enters state
* Donor enters zip code
* Donor enters country
* Donor enters email
* Donor enters phone number
* Donor enters amount to be donated
* Donor clicks submit
* Donation is logged
* Donation is processed

## Alternative Flows

### < First Alternative Flow >

* If the donor at any point hits the cancel button
* donor clicks okay when told changes will be lost
* No donation is logged.

# Special Requirements

## < First Special Requirement >

HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

# Pre-conditions

## < Pre-condition One >

Donors must be on the donation page on the HOOF website.

# Post-conditions

## < Post-condition One >

The details of the donation will have been logged in the database and the transaction will be processed.

# Use-Case Name

## Brief Description

This use case describes how a donor will schedule donations in the future.

# Flow of Events

## Basic Flow

* Donor enters amount to be donated
* Donor enters start date of future donation
* Donor enters end date of future donation
* Donor selects rate of recurrence of donation
* Donor clicks submit
* Donation details are logged
* Donation transaction is processed on the selected dates

## Alternative Flows

### < First Alternative Flow >

* If the donor at any point hits the cancel button
* donor clicks okay when told changes will be lost
* No donation is logged.

# Special Requirements

## < First Special Requirement >

HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

# Pre-conditions

## < Pre-condition One >

Donors must be on the donation page on the HOOF website, in the “Schedule a Future Donation” section.

## < Pre-condition Two >

Donor must be logged into their user account to schedule a future donation.

# Post-conditions

## < Post-condition One >

The details of the future donation will have been logged in the database.

## < Post-condition Two >

The donation transaction will be processed on the selected dates.

# Use-Case Name

## Brief Description

This use case describes how donor will cancel scheduled or recurring donation.

# Flow of Events

## Basic Flow

* Donor clicks “Cancel Scheduled Donation”
* Donor selects the start date of the cancelation
* Donor clicks submit

## Alternative Flows

### < First Alternative Flow >

* If the donor at any point hits the cancel button
* donor clicks okay when told changes will be lost
* No changes to the scheduled donations are logged.

# Special Requirements

## < First Special Requirement >

HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

# Pre-conditions

## < Pre-condition One >

Donors must be on the donation page on the HOOF website, in the “Schedule a Future Donation” section.

## < Pre-condition Two >

Donor must be logged into their user account to schedule a future donation.

# Post-conditions

## < Post-condition One >

Donations scheduled on/after selected cancelation date are canceled.

# Use-Case Name

## Brief Description

This use case describes how the donor receives a receipts for their donation.

# Flow of Events

## Basic Flow

* Donor makes a donation on HOOF KY website
* Receipt is automatically generated by HOOF’s system
* HOOF’s system triggers an automatic email to be sent to the email listed by the donor
* Donor receives receipt email for the donation they made

## Alternative Flows

### < First Alternative Flow >

* If the donor pays a donation via mailing a check
* HOOF employee will print a receipt for the donation
* HOOF will mail the receipt via mail to the return address listed by the donor
* Donor receives receipt

### < Second Alternative Flow >

* If the donor pays a scheduled donation
* Automatic receipt email is triggered
* Donor will receive a receipt via email

# Special Requirements

## < First Special Requirement >

HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

## < Second Special Requirement >

HOOF must have an automated emailing system able to send an email to the listed email with an attached copy of the donation receipt.

# Pre-conditions

## < Pre-condition One >

Donor must have access to the email account they entered when making their donation.

# Post-conditions

## < Post-condition One >

The donation receipt will be logged in HOOF KY’s database.

# Use-Case Name

## Brief Description

This use case describes how the donor receives a thank you for their donation.

# Flow of Events

## Basic Flow

* Donor makes a donation on HOOF KY website
* Thank You email is automatically generated by HOOF’s system
* HOOF’s system triggers an automatic email to be sent to the email listed by the donor
* Donor receives Thank You email for the donation they made

## Alternative Flows

### < First Alternative Flow >

* If the donor pays a donation via mailing a check
* HOOF employee will print a Thank You message for the donation
* HOOF will mail the Thank You message via mail to the return address listed by the donor
* Donor receives Thank You message

### < Second Alternative Flow >

* If the donor pays a scheduled donation
* Automatic Thank You email is triggered
* Donor will receive a Thank You message via email

# Special Requirements

## < First Special Requirement >

HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

## < Second Special Requirement >

HOOF must have an automated emailing system able to send an email to the listed email with a Thank You message in the email.

# Pre-conditions

## < Pre-condition One >

Donor must have access to the email account they entered when making their donation.

# Post-conditions

## < Post-condition One >

The confirmation of a Thank You message being sent to the donor will be logged in HOOF KY’s database.

# Use-Case Name

## Brief Description

This use case describes how the user will sign up for the HOOF KY newsletter.

# Flow of Events

## Basic Flow

* User clicks “Sign up For Our Newsletter”
* User is asked to confirm their email attached to their account is correct
* User clicks “Confirm”
* User is subscribed to the Newsletter
* User’s account is added to the Newsletter address book

## Alternative Flows

### < First Alternative Flow >

* If the user at any point hits the cancel button
* User clicks okay when told changes will be lost
* No Newsletter subscription is logged.

# Special Requirements

## < First Special Requirement >

HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

# Pre-conditions

## < Pre-condition One >

User must be on the HOOF website.

## < Pre-condition Two >

User must be logged into their user account to sign up for the newsletter.

# Post-conditions

## < Post-condition One >

The user will receive future sends of the HOOF newsletter.

# Use-Case Name

## Brief Description

This use case describes how the user will unsubscribe from the HOOF KY newsletter.

# Flow of Events

## Basic Flow

* User clicks on “Manage my Newsletter subscription”
* User selects to unsubscribe from the Newsletter
* User clicks “Confirm”
* User is unsubscribed to the Newsletter
* User’s account is removed from the Newsletter address book

## Alternative Flows

### < First Alternative Flow >

* If the user at any point hits the cancel button
* User clicks okay when told changes will be lost
* No Newsletter subscription is logged.

### < Second Alternative Flow >

* If the user wants to unsubscribe through a newsletter email
* User clicks unsubscribe within the email
* User is redirected to the HOOF website
* User is notified they have been unsubscribed
* User is removed from the newsletter address book

# Special Requirements

## < First Special Requirement >

HOOF must have an applicable database in which it is able to store all of its relevant information such as the items listed within the main flow.

# Pre-conditions

## < Pre-condition One >

User must be on the HOOF website.

## < Pre-condition Two >

User must be logged into their user account to sign up for the newsletter.

# Post-conditions

## < Post-condition One >

The user will not receive future sends of the HOOF newsletter.

Use Case Specification 30: User Follows HOOF KY on Social Media

1. **Brief Description**  
   Describes how donor will follow HOOF KY on Social Media  
    **Primary Actor**: Donor

1. **Flow of Events**

2.1 **Main Flow**:

* The use case starts with the Donor looking at HOOF KY’s website
* The donor clicks on the Facebook link under the “Contact” category
* The donor logs into their Facebook account
* The donor hits the follow link on the HOOF KY Facebook page
* The use case ends

2.2 **Alternative Flow**: None

1. **Pre conditions**:

* Donor already has Facebook

1. **Post conditions**:

* Donor receives notification of the follow on Facebook

Use Case Specification 31: Volunteer Modifies Data

1. **Brief Description**: Describes how a Volunteer will modify their personal data  
    Primary Actor: Volunteer

1. **Flow of Event**:

6.1 **Main Flow**:

* The volunteer will enter their login information
* The volunteer will access their personal data file
* The volunteer will make modifications accordingly
* The volunteer database will update
* The volunteer will log out of profile
* The use case ends

6.2 **Alternative Flow**: None

1. **Pre conditions**:

* Volunteer has a personal laptop/device to access data
* Volunteer has already registered
* Volunteer database is established

1. **Post conditions**:

* Volunteers data will be changed
* Data will update automatically to database

Use Case Specification 32: Volunteer Deletes Data

1. **Brief Description**: Describes how the volunteer will delete personal data  
    **Primary Actor**: Volunteer

1. **Flow of Events**

10.1 **Main Flow**:

* The volunteer will enter their login information
* The volunteer will access their personal data file
* The volunteer will delete data accordingly
* The volunteer database will update
* The volunteer will log out of profile
* The use case ends

10.2 **Alternative Flow**: None

1. **Pre conditions**:

* Volunteer has a personal laptop/device to access data
* Volunteer has already registered
* Volunteer database is established

1. **Post conditions**:
2. Volunteers data will be changed
3. Data will update automatically to database

Use Case Specification 33: Volunteer is Assigned to an Event

1. **Brief Description**: Describes how the Volunteer will be assigned to an event  
    Primary Actor: Volunteer

1. **Flow of Events**

14.1 **Main Flow**:

* The HOOF Director in charge of event planning will log into the Event Database
* The HOOF Director will assign Volunteer(s) to specific events
* Email is automatically sent to verify Volunteer of which event they are assigned to
* The Volunteer is alerted what event they are assigned to.
* The Volunteer confirms notification
* The Use Case ends

14.2 **Alternative Flow**: None

1. **Pre conditions**:

* Volunteer is already accepted to be a volunteer

1. **Post conditions**:

* Event Database is updated
* Volunteer is notified of event

Use Case Specification 34: Volunteer is Reassigned to Event

1. **Brief Description**: Describes how the Volunteer will be reassigned to another event

**Primary Actor**: Volunteer

1. **Flow of Events**

18.1 **Main Flow**:

* The HOOF Director in charge of event planning will log into the Event Database
* The HOOF Director will move Volunteer from one event to another
* Email is automatically sent to verify Volunteer of which event they are reassigned to
* The Volunteer is alerted what event they are reassigned to
* The Volunteer confirms notification
* The Use Case ends

* 1. **Alternative Flow**: None

1. **Pre conditions**:

* Volunteer is already accepted to be a volunteer

1. **Post conditions**:

* Event Database is updated
* Volunteer is notified of event change

Use Case Specification 35: Volunteer is Dropped from Event

1. **Brief Description**: Describes how the Volunteer will be dropped from an event

**Primary Actor**: Volunteer

1. **Flow of Events**:

22.1 **Main Flow**:

* The HOOF Director in charge of event planning will log into the Event Database
* The HOOF Director will remove the Volunteer from an event
* Email is automatically sent to verify Volunteer of which event they are dropped from
* The Volunteer is alerted what event they are dropped from
* The Volunteer confirms notification
* The Use Case ends

22.2 **Alternate Flow**: None

1. **Pre conditions**:

* Volunteer is already accepted to be a volunteer

1. **Post conditions**:

* Event Database is updated
* Volunteer is notified of event change

Use Case Specification 36: Volunteer Receives Confirmation

1. **Brief Description**: Describes how the Volunteer will receive Volunteer confirmation, contact info, and thanks for volunteering  
    **Primary Actor**: Volunteer

1. **Flow of Events**

26.**1 Main Flow**:

* The HOOF Directors in charge of the volunteer database and Event database will log in
* The HOOF Directors will add appropriate information into databases
* Email is automatically sent to Volunteer to give confirmation on said event(s) they are placed in
* The Volunteer confirms notification
* The HOOF Director in charge of the Event(s) will send thank you email to the Volunteer
* The Volunteer receives thank you email
* The Use case ends

26.2 **Alternative Flow**: None

1. **Pre conditions**:

* Volunteer is already accepted to be a volunteer

1. **Post conditions**:

* Volunteer will complete event

Use Case Specification 37: Volunteer Logs Hours

1. **Brief Description**: Describes how Volunteers will log their Volunteer Hours

**Primary Actor**: Volunteer

1. **Flow of Events**

30.1 **Main Flow**:

* The Volunteer will enter login information to access their personal profile within the Volunteer database
* The Volunteer will log their hours into the system
* The Volunteer Database will automatically update
* The Use Case ends

* 1. **Alternative Flow**: None

1. **Pre conditions**:

* Volunteer is already accepted to be a volunteer
* Volunteer has participated in event

1. **Post conditions**:

* Volunteer hours are logged and updated in Volunteer Database

Use Case Specification 38: Volunteer Deletes Log Hours

1. **Brief Description**: Describes how Volunteers will delete their Volunteer Hours

**Primary Actor**: Volunteer

1. **Flow of Events**

30.1 **Main Flow**:

* Volunteer selects the hours from their timesheet the wish to delete
* Volunteer clicks delete button
* Volunteer clicks the confirm button when prompted.
* Log item is deleted

* 1. **Alternative Flow**:
     1. Volunteer hits cancel at any point
     2. The log item is not deleted

1. **Pre conditions**:

* Volunteer is already accepted to be a volunteer
* Volunteer has participated in event

1. **Post conditions**:

* Volunteer hours are deleted and removed from Volunteer Database

Use Case Specification 39: Volunteer Modifies Log Hours

1. **Brief Description**: Describes how Volunteers will modify their Volunteer Hours

**Primary Actor**: Volunteer

1. **Flow of Events**

30.1 **Main Flow**:

* Volunteer selects the hours from their timesheet the wish to modify
* Volunteer selects new date if needed
* Volunteer selects new hours if needed\
* Volunteer clicks confirm
* Lod item is updated

* 1. **Alternative Flow**:
     1. Volunteer hits cancel at any point
     2. The log item is not updated.

1. **Pre conditions**:

* Volunteer is already accepted to be a volunteer
* Volunteer has participated in event

1. **Post conditions**:

* Volunteer hours are updated and changed in Volunteer Database

Use Case Specification 40: Camper Signs Up

# Use-Case Name: Camper Signs Up

## Brief Description

This use case covers the events that occur when a camper seeks to sign up for the program. The parent, guardian, or teacher will fill out the form proposing the candidate to HOOF, which will pass to HOOF’s system for further evaluation by HOOF staff.

# Flow of Events

## Basic Flow

* Nominator enters Child’s Name
* Nominator Enters Child’s Address Line 1
* Nominator Enters Child’s city
* Nominator enters child’s state
* Nominator enters child’s zip
* Nominator clicks confirm details
* Registration created

## Alternative Flows

### User Cancels Registration

* The user may cancel entering their information at any point during the main flow or if the process times out due to inactivity.
* The user will be asked to confirm to make sure the user has not accidently clicked the cancel button and to give a chance to avoid the timeout.
* In the case the user does not avoid cancellation, the user’s session will be terminated, and no entry submitted for review by HOOF.

# Special Requirements

## < First Special Requirement >

This Process will be in the form of a web page, tied into the database. This should allow the system to compatible with a wide variety of operating systems and hardware configurations.

# Pre-conditions

## < Pre-condition One >

The database must be ready to accept new entries, and the web server must be online and operational. The user must have access to all the information the system will ask for to successfully complete the registration process. The parent must be on HOOF’s web page and have hit the sign up button

# Post-conditions

## < Post-condition One >

The system will have new potential camper information should the user successfully register themselves for the system.

Use Case Specification 41: Camper Cancels Camp Event

# Use-Case Name

## Brief Description

This use case covers the process of a camper cancelling their attendance to a camp event they are scheduled to participate in.

# Flow of Events

## Basic Flow

* The parent/guardian will click the cancel registration button
* The parent/guardian will click yes when prompted to confirm if they wish to cancel the registration
* The parent/guardian will be returned to the screen of events for their child
* Event registration removed.

## Alternative Flows

### < First Alternative Flow >

* parent/guardian clicks cancel button on the cancellation prompt.
* parent/guardian returned to their even registration
* Registration still exists.

# Special Requirements

## < First Special Requirement >

# Pre-conditions

## < Pre-condition One >

The user must be the parent or legal guardian of an attendee of a HOOF camp event. This attendee must already be scheduled to participate in a HOOF camp event. The parent/guardian will be navigated to the events page for their child.

# Post-conditions

## < Post-condition One >

The system now has a free slot from the user’s cancellation. This will allow another camper from the wait list to be entered into the program. The user’s entry will be cancelled but their details will be kept in the system such that they may attend camp events in the future.

## < Post-condition Two >

The camper does not continue the process of cancelling their registration. The registration will remain in place and the camper will still be expected to attend the event. The user will be able to cancel their event in the future should they still wish to do so.

Use Case Specification 42: Board Member Uploads Files

# Use-Case Name

## Brief Description

This use case covers the how a board member will upload files to share with other board members of HOOF.

# Flow of Events

## Basic Flow

* Board member enter file name to upload.
* Board member will click upload file
* Board member selects file location in pup up dialogue
* Board member enters additional details of file.
* Board member selects whether file will be read only
* Board member clicks confirm upload button

## Alternative Flows

### < First Alternative Flow >

* User Clicks cancel upload button
* User Closes page

# Special Requirements

## < First Special Requirement >

User must have a file ready to upload. The file should be in a pre-approved format to be uploaded to facilitate compatibility.

# Pre-conditions

## < Pre-condition One >

The board member must have a pre-created administrator account that will give them access to the administrator section of the website. They must access this section of the website before they can upload files. They have clicked the upload file section.

# Post-conditions

## < Post-condition One >

The system now has the file uploaded to the Organization’s file sharing. All the HOOF board member will be able to access this file and modify it unless it was locked by the uploading member

## < Post-condition Two >

The file was not uploaded, and the file sharing will feature no new files on the system.

Use Case Specification 43: Board Member Modifies Files

# Use-Case Name

## Brief Description

This use case covers how a board member may modify a file that has been uploaded to the HOOF board file exchange.

# Flow of Events

## Basic Flow

* Board member will select file of their choice
* Board member clicks edit button
* Board member will be able to edit the file
* Board member confirms edits
* Board member updates name
* Board member updates Description
* Board member Clicks Confirm Button
* File is updated.

## Alternative Flows

### < First Alternative Flow >

* If the uploader has marked the file as read only, Board member will be shown error message
* Board Member will click okay
* Board member will not edit file.

# Special Requirements

## < First Special Requirement >

Files must have been uploaded to the file exchange in order for there to be files to edit. The board member must have the editing software needed for the given file type.

# Pre-conditions

The board member must be in the shared file section of the HOOF board section.

## < Pre-condition One >

# Post-conditions

## < Post-condition One >

The existing file on the database has been modified. The new information will be available for all of the board member to access.

## < Post-condition Two >

The file will not have been modified. The original file will still be available to the board members to view.

Use Case Specification 44: Board Member Deletes Files

# Use-Case Name

## Brief Description

This use case covers how a board member may delete a file that has already been uploaded to the file sharing system of HOOF.

# Flow of Events

## Basic Flow

* Board member will select file of their choice
* Board member clicks delete button
* User clicks okay when confirmation to delete file is asked
* User is sent back to file exchange
* File is removed.

## Alternative Flows

### < First Alternative Flow >

* If the uploader has marked the file as read only, Board member will be shown error message
* Board Member will click okay
* Board member will not delete file.

# Special Requirements

## < First Special Requirement >

Files must have been uploaded to the file exchange in order for there to be files to delete. The board member must have the privileges needed to delete a given file.

# Pre-conditions

The board member must be in the shared file section of the HOOF board section.

## < Pre-condition One >

# Post-conditions

## < Post-condition One >

The existing file on the database has been deleted. The file will no longer be available for use by the other board members of HOOF.

## < Post-condition Two >

The file will not have been deleted. The original file will still be available to the board members to view.

Use Case Specification 45: Modify Auction Item

# Use-Case Name

## Brief Description

This use case covers how a volunteer may update the details for an item that is registered for auction at a HOOF event.

# Flow of Events

## Basic Flow

* Volunteer clicks item of their choice from the item list
* Volunteer clicks modify button
* Volunteer updates name (if needed)
* Volunteer updates description (if needed)
* Volunteer updates reserve amount (if needed)
* Volunteer hits confirm changes button
* Item is updated

## Alternative Flows

### < First Alternative Flow >

* If the volunteer at any point hits the cancel button
* Volunteer clicks okay when told changes will be lost
* Item remains unchanged.

# Special Requirements

## < First Special Requirement >

An auction must have been scheduled for the HOOF organization.

# Pre-conditions

Volunteers must be on the event’s page on the HOOF website, in the section for item registration.

## < Pre-condition One >

# Post-conditions

## < Post-condition One >

The details of the auction item will have been updated. The new details are available for other volunteers to view.

## < Post-condition Two >

The information will not have been updated. The old information will still be available for the other volunteers to see.

Use Case Specification 46: Delete Auction Item

# Use-Case Name

## Brief Description

This use case covers how a donor may delete an item that has been registered for HOOF’s auction events.

# Flow of Events

## Basic Flow

* Donor clicks item of their choice from the item list
* Donor clicks delete button
* Donor clicks okay on confirm deletion dialogue
* Donor is returned to event page
* Item is deleted

## Alternative Flows

### < First Alternative Flow >

* If the donor at any point hits the cancel button
* Volunteer clicks okay when told changes will be lost
* Item remains unchanged.

# Special Requirements

## < First Special Requirement >

An auction must have been scheduled for the HOOF organization.

# Pre-conditions

Volunteers must be on the event’s page on the HOOF website, in the section for item registration.

## < Pre-condition One >

# Post-conditions

## < Post-condition One >

The auction item will be deleted from the event. It will no longer be available for volunteers to view.

## < Post-condition Two >

The information will not have been deleted. The information will still be available for the volunteers to see.