ELABORATION PHASE SPECIFICATION

LOUISVILLE BUSINESS ANALYSIS

Bradley Erickson
Deshan Gardner
Chris James
Spencer Kerber
# TABLE OF CONTENTS

## Contents

Executive Summary .......................................................... 1
System Requirements .......................................................... 2
Trace Matrix ................................................................. 4
Use Case Diagram ........................................................... 6
Use Cases: Add Student ......................................................... 7
Use Cases: Modify Student ..................................................... 8
Use Cases: Delete Student ..................................................... 9
Use Cases: View Courses ....................................................... 10
Use Cases: View Course ........................................................ 11
Use Cases: Enroll Student ....................................................... 12
Use Cases: Review Course ...................................................... 13
Use Cases: Log In Student ....................................................... 14
Use Cases: Add Donation ......................................................... 15
Use Cases: Add Donor .......................................................... 16
Use Cases: Add Recurring Donation ........................................... 17
Use Cases: Modify Donor ........................................................ 18
Use Cases: Delete Donor ........................................................ 19
Use Cases: View Projects ....................................................... 20
Use Cases: View Donations ...................................................... 21
Use Cases: Log In Donor ......................................................... 22
Use Cases: Add Volunteer ......................................................... 23
Use Cases: Modify Volunteer .................................................... 24
Use Cases: Delete Volunteer ..................................................... 25
Use Cases: View Events ........................................................ 26
Use Cases: Register Volunteer .................................................. 27
Use Cases: Log In Volunteer ...................................................... 28
<table>
<thead>
<tr>
<th>Use Cases</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Teacher</td>
<td>29</td>
</tr>
<tr>
<td>Modify Teacher</td>
<td>31</td>
</tr>
<tr>
<td>Delete Teacher</td>
<td>32</td>
</tr>
<tr>
<td>Add Course</td>
<td>33</td>
</tr>
<tr>
<td>Modify Course</td>
<td>35</td>
</tr>
<tr>
<td>Delete Course</td>
<td>36</td>
</tr>
<tr>
<td>Log In Teacher</td>
<td>37</td>
</tr>
<tr>
<td>Add Photo</td>
<td>38</td>
</tr>
<tr>
<td>Modify Photo</td>
<td>39</td>
</tr>
<tr>
<td>Delete Photo</td>
<td>41</td>
</tr>
<tr>
<td>Add Event</td>
<td>42</td>
</tr>
<tr>
<td>Modify Event</td>
<td>43</td>
</tr>
<tr>
<td>Delete Event</td>
<td>44</td>
</tr>
<tr>
<td>View Data</td>
<td>45</td>
</tr>
<tr>
<td>Create Grant Report</td>
<td>46</td>
</tr>
<tr>
<td>Log In Administrator</td>
<td>47</td>
</tr>
<tr>
<td>View Events</td>
<td>48</td>
</tr>
<tr>
<td>View Event</td>
<td>49</td>
</tr>
<tr>
<td>Purchase Ticket</td>
<td>50</td>
</tr>
<tr>
<td>Class Diagram</td>
<td>51</td>
</tr>
<tr>
<td>Data Dictionary</td>
<td>52</td>
</tr>
<tr>
<td>Student</td>
<td>53</td>
</tr>
<tr>
<td>Donor</td>
<td>54</td>
</tr>
<tr>
<td>Volunteer</td>
<td>55</td>
</tr>
<tr>
<td>Teacher</td>
<td>56</td>
</tr>
<tr>
<td>Administrator</td>
<td>57</td>
</tr>
<tr>
<td>User</td>
<td>58</td>
</tr>
<tr>
<td>Session</td>
<td>59</td>
</tr>
<tr>
<td>CONTENT</td>
<td>PAGE</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>Course</td>
<td>60</td>
</tr>
<tr>
<td>Event</td>
<td>61</td>
</tr>
<tr>
<td>Donation</td>
<td>62</td>
</tr>
<tr>
<td>Photo</td>
<td>63</td>
</tr>
<tr>
<td>Grant Report</td>
<td>64</td>
</tr>
<tr>
<td>Survey</td>
<td>65</td>
</tr>
<tr>
<td>Stripe Payment</td>
<td>66</td>
</tr>
<tr>
<td>Windows Navigation Diagrams</td>
<td>67</td>
</tr>
<tr>
<td>Physical Architecture Design</td>
<td>68</td>
</tr>
<tr>
<td>Design Procedures for Security</td>
<td>69</td>
</tr>
<tr>
<td>Gantt Chart</td>
<td>70</td>
</tr>
<tr>
<td>Prototypes</td>
<td>71</td>
</tr>
<tr>
<td>Main Page</td>
<td>72</td>
</tr>
<tr>
<td>View Classes/Events Page</td>
<td>73</td>
</tr>
<tr>
<td>View Class/Event</td>
<td>74</td>
</tr>
<tr>
<td>Register for Class/Event page</td>
<td>75</td>
</tr>
<tr>
<td>Donate page</td>
<td>76</td>
</tr>
<tr>
<td>Donation Successful Page</td>
<td>77</td>
</tr>
<tr>
<td>LOLA Lovers Landing Page</td>
<td>78</td>
</tr>
<tr>
<td>Volunteer Login Page</td>
<td>79</td>
</tr>
<tr>
<td>Volunteer View Event Page</td>
<td>80</td>
</tr>
<tr>
<td>Volunteer Registration Page</td>
<td>81</td>
</tr>
<tr>
<td>Modify Volunteer Account Page</td>
<td>82</td>
</tr>
<tr>
<td>Admin Login Page</td>
<td>83</td>
</tr>
<tr>
<td>Admin Dashboard Page</td>
<td>84</td>
</tr>
<tr>
<td>Grant Report Page</td>
<td>85</td>
</tr>
<tr>
<td>Create Admin Account Page</td>
<td>86</td>
</tr>
<tr>
<td>View All Accounts Page</td>
<td>87</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

View All Classes/Events page ................................................................. 88
Add Volunteer Page ............................................................................... 89
Add Teacher page .................................................................................. 90
Modify Admin Account Page ............................................................... 91
Create Event Page .................................................................................. 92
Edit Event Page ....................................................................................... 93
Login Teacher Page ................................................................................ 94
Teacher Landing Page ............................................................................ 95
Modify Teacher Page ............................................................................. 96
Purchase Event Ticket Page ................................................................. 97
Contact Information .............................................................................. 97
Company Information ............................................................................ 97
Executive Summary

OBJECTIVES
Our objective is to create a new Land O'Lakes website that will better enhance their online presence.

GOALS
Our goals are to update the areas where the current Land O'Lakes website is lacking and improve the current features used by the website.

SOLUTION
We hope to add a new student portal that will allow online student registration. We also hope to add both a teacher and volunteer portal so they can better manage courses and events. We hope to add a donation portal that will allow donors to create one-time and recurring donations. Finally, we hope to add a data collection portal where administrators will be able to gather data collected by the website and organize it into grant reports.
System Requirements

The system Requirements provides a hierarchy of both functional and nonfunctional requirements that the system will need in order to perform its designed functions. The requirements are separated into categories based on the functions primary user. Requirements in category 1 correspond to the overall website functionality. Requirements in category 2 correspond to functions involved in the donation process. Requirements in category 3 correspond to functions involving students, including class registration, tuition payment, and class reviews. The requirements in category 4 correspond to functions involving staff and volunteers, including information about teacher class assignments and volunteer event assignments. The requirements in category 5 involve the need for a grant report system. These include data collection and an admin portal to view collected data.
## System Requirements

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.0</strong></td>
<td>The website shall be designed in WordPress to be able to easily add, modify, and delete pages as needed</td>
</tr>
<tr>
<td><strong>2.0</strong></td>
<td>The website shall have a donation portal</td>
</tr>
<tr>
<td><strong>2.1</strong></td>
<td>Donations shall be made using Stripe</td>
</tr>
<tr>
<td><strong>2.2</strong></td>
<td>Donors shall have the ability to make both recurring and one time donations</td>
</tr>
<tr>
<td><strong>2.3</strong></td>
<td>Donation data shall be gathered</td>
</tr>
<tr>
<td><strong>2.4</strong></td>
<td>Donors shall have the ability to donate towards a specific project</td>
</tr>
<tr>
<td><strong>2.5</strong></td>
<td>Donors shall have the ability to see where their donations are used</td>
</tr>
<tr>
<td><strong>3.0</strong></td>
<td>The website shall have a student portal</td>
</tr>
<tr>
<td><strong>3.1</strong></td>
<td>Students shall have the ability to register for classes online</td>
</tr>
<tr>
<td><strong>3.2</strong></td>
<td>Students shall have the ability to pay for classes online</td>
</tr>
<tr>
<td><strong>3.3</strong></td>
<td>Students shall have the ability to post class reviews</td>
</tr>
<tr>
<td><strong>3.4</strong></td>
<td>Students shall have the ability to search for currently offered classes</td>
</tr>
<tr>
<td><strong>4.0</strong></td>
<td>The website shall have an staff portal</td>
</tr>
<tr>
<td><strong>4.1</strong></td>
<td>Staff shall have the ability to submit request to teach a class</td>
</tr>
<tr>
<td><strong>4.2</strong></td>
<td>Volunteers shall have the ability to sign up for specific events</td>
</tr>
<tr>
<td><strong>4.3</strong></td>
<td>Instructors shall be able to view class information about their classes</td>
</tr>
<tr>
<td><strong>5.0</strong></td>
<td>The website shall have the ability to produce grant reports</td>
</tr>
<tr>
<td><strong>5.1</strong></td>
<td>Staff shall be able to view grant reports</td>
</tr>
<tr>
<td><strong>5.2</strong></td>
<td>The website shall have the ability to collect and store data about business activities</td>
</tr>
</tbody>
</table>
Trace Matrix

The trace matrix matches each use case to what system requirement it is dependent on.

<table>
<thead>
<tr>
<th></th>
<th>1 A</th>
<th>1 B</th>
<th>1 C</th>
<th>1 D</th>
<th>1 E</th>
<th>1 F</th>
<th>1 G</th>
<th>1 H</th>
<th>2 A</th>
<th>2 B</th>
<th>2 C</th>
<th>2 D</th>
<th>2 E</th>
<th>2 F</th>
<th>2 G</th>
<th>2 H</th>
<th>3 A</th>
<th>3 B</th>
<th>3 C</th>
<th>3 D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The trace matrix matches each use case to what system requirement it is dependent on.
<table>
<thead>
<tr>
<th></th>
<th>3E</th>
<th>3F</th>
<th>4A</th>
<th>4B</th>
<th>4C</th>
<th>4D</th>
<th>4E</th>
<th>5A</th>
<th>5B</th>
<th>5C</th>
<th>5D</th>
<th>5E</th>
<th>5F</th>
<th>5G</th>
<th>5H</th>
<th>5I</th>
<th>5J</th>
<th>6A</th>
<th>6B</th>
<th>6C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Use Case Diagram

HOW TO READ THE USE CASE DIAGRAM

The use case diagram displays how different types of users interact with the system.
Use Cases: Add Student

ADD STUDENT
This use case describes how a student would create an account

USE CASE: ADD STUDENT

<table>
<thead>
<tr>
<th>Risk Level: Low</th>
<th>Preconditions: Student account does not exist</th>
<th>Postconditions: Student account is created</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID: 1A</td>
<td>Primary Actor: Student</td>
<td></td>
</tr>
</tbody>
</table>

Main Flow:
1. Visitor goes to www.lolaartswi.com
2. Visitor clicks on "Become a Student" button on the homepage.
3. Visitor is directed to Student account creation page.
4. Visitor enters first name in first name text box.
5. Visitor enters last name in last name text box.
6. Visitor enters email in email text box.
7. Visitor enters password in password text box.
8. Visitor enters password in re-enter password text box.
9. Visitor enters address in address text box.
10. Visitor enters city in city text box.
11. Visitor selects state from combo box.
12. Visitor enters zip code in zip code text box.
13. Visitor clicks "Become a Student" button at bottom of page.
14. Visitor is directed to Student account page.
15. Visitor is added to student table in database.
16. Visitor receives confirmation email.
17. Use case ends.
Use Cases: Modify Student

MODIFY STUDENT
This use case describes how a student would modify his or her account

USE CASE: MODIFY STUDENT
ID: 1B
Risk Level: Low
Preconditions: None
Postconditions: Student account is modified
Primary Actor: Student

Main Flow:
1. User goes to www.lolaartwsi.com
2. User Enter username in the textbox
3. User Enters password in the textbox
4. User Clicks Log on
5. User Clicks on Students
6. User Selects a student to modify
7. User then clicks modify Student
8. User modify student
9. User clicks save
10. A pop up display asking if you are sure you want to save changes
11. User Clicks “yes”
12. User then directed to new screen which says “Changes Saved”
13. Student is modified
14. Use Case Ends
Use Cases: Delete Student

DELETE STUDENT
This use case describes how a student would delete his or her account

USE CASE: DELETE STUDENT
ID: 1C

<table>
<thead>
<tr>
<th>Risk Level: Low</th>
<th>Primary Actor: Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions: None</td>
<td>Postconditions: Account is deleted</td>
</tr>
</tbody>
</table>

Main Flow:
1. User goes to www.lolaartwsi.com
2. User Enter username in the textbox
3. User Enters password in the textbox
4. User Clicks Log on
5. Admin directed to admin page
6. User Clicks on Students
7. User Selects a student to delete
8. User then clicks Delete Student
9. A pop up display asking if you are sure you want to delete student
10. User Clicks “Yes”
11. User then directed to new showing “Student Deleted successful”
12. Student is deleted
13. Use Case Ends
Use Cases: View Courses

VIEW COURSES
This use case describes how a student would view all available classes

USE CASE: VIEW CLASSES

<table>
<thead>
<tr>
<th>Risk Level: Low</th>
<th>ID: 1D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Actor: Student</td>
<td></td>
</tr>
</tbody>
</table>

Preconditions: None

Postconditions: Student is directed to classes page

Main Flow:
1. User goes to www.lolaartwsi.com
2. User Enter username in the textbox
3. User Enters password in the textbox
4. User Clicks “Log on”
5. Student Directed to student Page
6. Student clicks “Classes”
7. Student Views Classes that are offered
8. Use case ends

Diagram: (Diagram not provided in text but typically includes a flowchart or process diagram illustrating the steps of the Main Flow.)
Use Cases: View Course

VIEW COURSE
This use case describes how a student would view class information

USE CASE: VIEW COURSE

<table>
<thead>
<tr>
<th>ID: 1E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
</tr>
<tr>
<td>Primary Actor: Student</td>
</tr>
<tr>
<td>Preconditions: None</td>
</tr>
<tr>
<td>Postconditions: Student is directed to class page</td>
</tr>
</tbody>
</table>

Main Flow:
1. User goes to www.lolaartwsi.com
2. User enters username in the textbox
3. User enters password in the textbox
4. User clicks "Log on"
5. Student directed to student page
6. Student clicks "Classes"
7. Student clicks on Class
8. Student views class info of selected class
9. Use case ends
Use Cases: Enroll Student

ENROLL STUDENT
This use case describes how a student would enroll in a class

<table>
<thead>
<tr>
<th>USE CASE: ENROLL STUDENT</th>
<th>ID: 1F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: High</td>
<td>Primary Actor: Student</td>
</tr>
<tr>
<td>Preconditions: None</td>
<td>Postconditions: Student is enrolled</td>
</tr>
</tbody>
</table>

**Main Flow:**
1. User goes to www.lolaartwsi.com
2. User Enter username in the textbox
3. User Enter password in the textbox
4. User Clicks “Log on”
5. Student Directed to student Page
6. Student Clicks “Classes”
7. Student Views Classes that are offered
8. Student Clicks on Class
9. Student views information about selected class
10. Student Clicks “Enroll Class”
11. Student directed to payment page
12. Student select method of payment
13. Student Enter Credentials
14. Student Clicks “Pay For Class”
15. Student directed to page saying “Successfully Enrolled “
16. Use case ends
Use Cases: Review Course

REVIEW CLASS
This use case describes how a student would post a class review.

<table>
<thead>
<tr>
<th>USE CASE: REVIEW CLASS</th>
<th>ID: 1G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
<td>Primary Actor: Student</td>
</tr>
<tr>
<td>Preconditions: Student has taken class</td>
<td>Postconditions: Student posts review</td>
</tr>
</tbody>
</table>

Main Flow:
1. User goes to www.lolaartwsi.com
2. User Enter username in the textbox
3. User Enters password in the textbox
4. User Clicks “Log on”
5. Student Directed to student Page
6. Student clicks “class reviews”
7. Student clicks on class
8. Student types review for selected class
9. Student then clicks “Submit Review”
10. Student then is directed to page which says “Review Successfully Submitted”
11. Use case ends
Use Cases: Log In Student

LOG IN
This use case describes how a student would log into his or her account

<table>
<thead>
<tr>
<th>USE CASE: LOG IN</th>
<th>ID: 1H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
<td>Primary Actor: Student</td>
</tr>
<tr>
<td>Preconditions: Student is not logged in</td>
<td>Postconditions: Student is logged in</td>
</tr>
</tbody>
</table>

Main Flow:
1. User goes to www.lolaartwsi.com
2. User Enter username in the textbox
3. User Enters password in the textbox
4. User Clicks “Log on”
5. Student Directed to student Page
6. Use case ends
Use Cases: Add Donation

ADD DONATION
This use case describes how a user would make a one time donation

<table>
<thead>
<tr>
<th>USE CASE: ADD DONATION</th>
<th>ID: 2A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: High</td>
<td>Primary Actor: Donor</td>
</tr>
<tr>
<td>Preconditions: None</td>
<td>Postconditions: Donation is made</td>
</tr>
<tr>
<td>Main Flow:</td>
<td></td>
</tr>
<tr>
<td>1. User goes to <a href="http://www.lolaartwsi.com">www.lolaartwsi.com</a></td>
<td></td>
</tr>
<tr>
<td>2. User enters username in user name textbox</td>
<td></td>
</tr>
<tr>
<td>3. User enters password in password textbox</td>
<td></td>
</tr>
<tr>
<td>4. User clicks log in button</td>
<td></td>
</tr>
<tr>
<td>5. User clicks on the donation tab from the home page</td>
<td></td>
</tr>
<tr>
<td>6. User is directed to the donation page</td>
<td></td>
</tr>
<tr>
<td>7. User enters needed information into given textboxes</td>
<td></td>
</tr>
<tr>
<td>8. User then selects if he/she wants a recurring donation or a one time donation</td>
<td></td>
</tr>
<tr>
<td>9. User then has option to select a given project or a general donation</td>
<td></td>
</tr>
<tr>
<td>10. User enters amount he/she wants to donate</td>
<td></td>
</tr>
<tr>
<td>11. User selects payment method (Credit card or pay-pal)</td>
<td></td>
</tr>
<tr>
<td>12. User enters credit card information if credit card is selected</td>
<td></td>
</tr>
<tr>
<td>13. User hits submit</td>
<td></td>
</tr>
<tr>
<td>14. New page pops up confirming donation</td>
<td></td>
</tr>
<tr>
<td>15. Use Case ends</td>
<td></td>
</tr>
</tbody>
</table>
Use Cases: Add Donor

ADD DONOR
This use case describes how a donor would make an account

USE CASE: ADD DONOR
ID: 2B
Risk Level: Low
Preconditions: None
Primary Actor: Donor
Postconditions: Donor account is created

Main Flow:
1. Visitor goes to www.lolaartswi.com
2. Visitor clicks on "Become a Donor" button on the homepage.
3. Visitor is directed to Donor account creation page.
4. Visitor enters first name in first name text box.
5. Visitor enters last name in last name text box.
6. Visitor enters email in email textbox.
7. Visitor enters password in password textbox.
8. Visitor enters password in re-enter password textbox.
9. Visitor enters address in address textbox.
10. Visitor enters city in city textbox.
11. Visitor selects state from combo box.
12. Visitor enters zip code in zip code textbox.
13. Visitor clicks "Become a Donor" button at bottom of page.
14. Visitor is directed to Donor account page.
15. Visitor is added to donor table in database.
16. Visitor receives confirmation email.
17. Use case ends.
Use Cases: Add Recurring Donation

ADD RECURRING DONATION
This use case describes how a donor would create a recurring donation

<table>
<thead>
<tr>
<th>USE CASE: ADD RECURRING DONATION</th>
<th>ID: 2C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: High</td>
<td>Primary Actor: Donor</td>
</tr>
<tr>
<td>Preconditions: None</td>
<td>Postconditions: Recurring donation is made</td>
</tr>
</tbody>
</table>

**Main Flow:**
1. User goes to www.lolaartwsi.com
2. User enters username in username textbox
3. User enters password in password textbox
4. User clicks log in button
5. User clicks on the donation tab from the home page
6. User is directed to the donation page
7. User enters needed information into given textboxes
8. User then selects if he/she wants a recurring donation or a one-time donation
9. User then has option to select a given project or a general donation
10. User enters amount he/she wants to donate
11. User selects recurring donation
12. User enters recursion period
13. User selects payment method (Credit card or pay-pal)
14. User enters credit card information if credit card is selected
15. User hits submit
16. New page pops up confirming donation
17. Use Case ends
Use Cases: Modify Donor

MODIFY DONOR
This use case describes how a donor would modify his or her account

USE CASE: MODIFY DONOR
ID: 2D
Risk Level: Low
Primary Actor: Donor
Preconditions: None
Postconditions: Donor account is modified

Main Flow:
1. User goes to www.lolaartswi.com
2. User then enters username into username textbox
3. User enters password into password textbox
4. User clicks log in
5. User is logged in
6. User clicks donation tab
7. User is directed to donation page
8. User clicks My account
9. User is directed to donation history page
10. User chooses which donation to edit
11. User clicks “Edit” next to selected donation
12. User edits donation by changing amount or changing recurring to one-time
13. User clicks “Save Changes”
14. User is redirected to confirmation page
15. Use Case Ends

User
Clicks "Edit" Next to selected donation
Request to edit
Request granted
edit chosen info
Hit “Save and Process”
Check for errors
No Errors
Successfully processed

Modify Donation
LOLA Website
Use Cases: Delete Donor

DELETE DONOR
This use case describes how a donor would delete his or her account

USE CASE: DELETE DONOR

ID: 2E

Risk Level: Low

Primary Actor: Donor

Preconditions: None

Postconditions: Donor account is deleted

Main Flow:
1. User goes to www.lolaartswi.com
2. User enters Username in Username textbox
3. User enters password into password textbox
4. User clicks “Log In”
5. User is logged in
6. User clicks on the donation tab
7. User is directed to the donation page
8. User clicks view donations
9. User is directed to current/past donations page
10. User clicks “Edit” next to donation wanted to be deleted
11. User is directed to edit donations page
12. User clicks “Delete Donation”
13. Message appears asking if user is sure he/she wants to delete donation
14. User clicks yes
15. User is directed to confirmation page
16. Use case ends
Use Cases: View Projects

VIEW PROJECTS
This use case describes how a donor would view projects accepting donations

<table>
<thead>
<tr>
<th>USE CASE: VIEW PROJECTS</th>
<th>ID: 2F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
<td>Primary Actor: Donor</td>
</tr>
<tr>
<td>Preconditions: None</td>
<td>Postconditions: Donor is directed to projects page</td>
</tr>
</tbody>
</table>

Main Flow:
1. User goes to www.lolaartswi.com
2. User clicks on donations tab
3. User is directed to donations page
4. Use case ends
Use Cases: View Donations

VIEW DONATIONS
This use case describes how a donor would view where his or her donations have been used

<table>
<thead>
<tr>
<th>USE CASE: VIEW DONATIONS</th>
<th>ID: 2F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
<td></td>
</tr>
<tr>
<td>Primary Actor: Donor</td>
<td></td>
</tr>
<tr>
<td>Preconditions: None</td>
<td>Postconditions: Donor is directed to donations page</td>
</tr>
</tbody>
</table>

Main Flow:
1. User goes to www.lolaartswi.com
2. User then enters username into username textbox
3. User enters password into password textbox
4. User clicks log in
5. User is logged in
6. User clicks donation tab
7. User is directed to donation page
8. User clicks My account
9. User is directed to donation history page
10. Use case ends
Use Cases: Log In Donor

LOG IN
This use case describes how a donor would log into his or her account

<table>
<thead>
<tr>
<th>USE CASE: LOG IN</th>
<th>ID: 2G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
<td>Primary Actor: Donor</td>
</tr>
<tr>
<td>Preconditions: Donor is not logged in</td>
<td>Postconditions: Donor is logged in</td>
</tr>
</tbody>
</table>

Main Flow:
1. User goes to www.lolaartswi.com
2. User then enters username into username textbox
3. User enters password into password textbox
4. User clicks log in
5. User is logged in
Use Cases: Add Volunteer

ADD VOLUNTEER
This use case describes how a user would create a volunteer account

<table>
<thead>
<tr>
<th>USE CASE: ADD VOLUNTEER</th>
<th>ID: 3A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
<td></td>
</tr>
<tr>
<td>Preconditions: None</td>
<td></td>
</tr>
<tr>
<td>Postconditions: Volunteer account is created</td>
<td></td>
</tr>
<tr>
<td>Primary Actor: User</td>
<td></td>
</tr>
</tbody>
</table>

**Main Flow:**
1. Visitor goes to www.lolaartswi.com
2. Visitor clicks on "Become a volunteer" button on the homepage.
3. Visitor is directed to volunteer account creation page.
4. Visitor enters first name in first name text box.
5. Visitor enters last name in last name text box.
6. Visitor enters email in email text box.
7. Visitor enters password in password text box.
8. Visitor enters password in re-enter password text box.
9. Visitor enters address in address text box.
10. Visitor enters city in city text box.
11. Visitor selects state from combo box.
12. Visitor enters zip code in zip code text box.
13. Visitor clicks "Become a volunteer" button at bottom of page.
14. Visitor is directed to volunteer account page.
15. Visitor is added to volunteer table in database.
16. Visitor receives confirmation email.
17. Use case ends.
Use Cases: Modify Volunteer

MODIFY VOLUNTEER
This use case describes how a volunteer would modify his or her account

USE CASE: MODIFY VOLUNTEER

<table>
<thead>
<tr>
<th>Risk Level: Low</th>
<th>Primary Actor: Volunteer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions: None</td>
<td>Postconditions: Volunteer account is modified</td>
</tr>
</tbody>
</table>

Main Flow:
1. Volunteer goes to www.lolaartswi.com
2. Volunteer clicks on volunteer tab on home page.
3. Volunteer enters username in username textbox.
4. Volunteer enters password in password textbox.
5. Volunteer clicks "log in" button
6. Volunteer is directed to account page.
7. Volunteer clicks "edit account" link.
8. Volunteer is directed to edit account page.
9. Volunteer changes account info
10. Volunteer clicks "save changes"
11. Volunteer is directed back to account page
12. New volunteer data is updated in database
13. Use case ends.
Use Cases: Delete Volunteer

DELETE VOLUNTEER
This use case describes how a volunteer would delete his or her account

USE CASE: DELETE VOLUNTEER
ID: 3C
Risk Level: Low
Primary Actor: Volunteer

Preconditions: None
Postconditions: Volunteer account is deleted

Main Flow:
1. Volunteer goes to www.lolaartswi.com
2. Volunteer clicks on volunteer tab on home page.
3. Volunteer enters username in username textbox.
4. Volunteer enters password in password textbox.
5. Volunteer clicks "log in" button
6. Volunteer is directed to account page.
7. Volunteer clicks on delete account link
8. Volunteer is directed to account deletion page
9. Volunteer clicks delete account.
10. Message box asking "are you sure you want to delete your account?"
11. User clicks "yes button"
12. Visitor is directed to successful account delete page.
13. Visitor receives confirmation email
14. Visitor account is removed from database.
15. Use case ends.
Use Cases: View Events

VIEW EVENTS
This use case describes how a user would view events accepting volunteers

USE CASE: VIEW EVENTS

<table>
<thead>
<tr>
<th>ID: 3D</th>
</tr>
</thead>
</table>

Risk Level: Low

Primary Actor: Volunteer

Preconditions: None

Postconditions: Volunteer is directed to events page

Main Flow:
1. Volunteer goes to www.lolaartswi.com
2. Volunteer clicks on volunteer tab on home page.
3. Volunteer enters username in username textbox.
4. Volunteer enters password in password textbox.
5. Volunteer clicks "log in" button
6. Volunteer is directed to volunteer account page
7. Volunteer clicks "view events" button
8. Volunteer is directed to volunteerable events page.
9. Use case ends
Use Cases: Register Volunteer

REGISTER VOLUNTEER
This use case describes how a volunteer would register for an event

<table>
<thead>
<tr>
<th>USE CASE: REGISTER VOLUNTEER</th>
<th>ID: 3E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
<td></td>
</tr>
<tr>
<td>Primary Actor: Volunteer</td>
<td></td>
</tr>
<tr>
<td>Preconditions: None</td>
<td></td>
</tr>
<tr>
<td>Postconditions: Volunteer is registered for event</td>
<td></td>
</tr>
</tbody>
</table>

Main Flow:
1. Volunteer goes to www.lolaartswi.com
2. Volunteer clicks on volunteer tab on home page.
3. Volunteer enters username in username textbox.
4. Volunteer enters password in password textbox.
5. Volunteer clicks "log in" button
6. Volunteer is directed to volunteer account page
7. Volunteer clicks "view events" button
8. Volunteer is directed to volunteerable events page.
9. Volunteer clicks desired event link
10. Volunteer is directed to event page
11. Volunteer clicks "volunteer for this event" button.
12. Message box "Are you sure you want to volunteer?"
13. Volunteer clicks "Yes" button
14. Volunteer is directed to event page.
15. Volunteer receives confirmation email
16. Event data in database is updated (volunteers needed is reduced by 1)
17. Use case ends

Volunteer

- findEvent
- addVolunteer

Events Accepting Volunteers

- searchEvents

Event Page

- addVolunteer

Volunteer Account Page
Use Cases: Log In Volunteer

LOG IN
This use case describes how a volunteer would log into his or her account

**USE CASE: LOG IN**

<table>
<thead>
<tr>
<th>Risk Level: Low</th>
<th>Primary Actor: Volunteer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions: Volunteer is not logged into account</td>
<td>Postconditions: Volunteer is logged into account</td>
</tr>
</tbody>
</table>

**Main Flow:**
1. Volunteer goes to www.lolaartswi.com
2. Volunteer clicks on volunteer tab on home page.
3. Volunteer enters username in username textbox.
4. Volunteer enters password in password textbox.
5. Volunteer clicks "log in" button
6. Volunteer is directed to volunteer account page
7. Use case ends
Use Cases: Add Teacher

ADD TEACHER
This use case describes how an administrator would create a teacher account.

USE CASE: ADD TEACHER

ID: 4A

Risk Level: Low
Primary Actor: Admin

Preconditions: Teacher does not have account
Postconditions: Teacher account is created

Main Flow:
1. Administrator goes to www.lolaartswi.com
2. Admin clicks "Administration" link at bottom of home page
3. Admin is directed to admin log in page
4. Admin enters username in username textbox
5. Admin enters password in password textbox
6. Admin clicks "log in" link
7. Admin is directed to Administration page
8. Admin clicks "Add instructor account" link
9. Admin is directed to instructor account creation page
10. Admin enters new instructor first name in first name textbox
11. Admin enters new instructor last name in last name textbox.
12. Admin enters new instructor address in address textbox.
13. Admin enters new instructor city in city textbox
14. Admin selects new instructor state from state combo box
15. Admin enters new instructor zip code in zip code textbox
16. Admin enters new instructor email in email textbox
17. Admin enters new instructor password in password textbox
18. Admin reenters new instructor password in password textbox
19. Admin enters new instructor phone number in phone number textbox
20. Admin enters new instructor class in class textbox.
21. Admin clicks "create instructor account" link
22. Admin is directed to successful account creation page "Creation successful"
23. New instructor data is added to database.
24. Instructor receives new account email confirmation
25. Use case ends
The LOLA administrator selects “add teacher”

The system creates a new teacher account

addTeacher(“UML”)

`<<Create>>`

UML: Teacher
Use Cases: Modify Teacher

MODIFY TEACHER

This use case describes how a teacher would modify his or her account.

<table>
<thead>
<tr>
<th>USE CASE: MODIFY TEACHER</th>
<th>ID: 4B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Risk Level:</strong> Low</td>
<td><strong>Primary Actor:</strong> Teacher</td>
</tr>
<tr>
<td><strong>Preconditions:</strong> None</td>
<td><strong>Postconditions:</strong> Teacher account is modified</td>
</tr>
</tbody>
</table>

**Main Flow:**
1. Instructor goes to www.lolaartswi.com
2. Instructor clicks "instructors" link at bottom of home page
3. Instructor is directed to instructor log in page
4. Instructor enters username in username text box
5. Instructor enters password in password textbox
6. Instructor clicks "log in"
7. Instructor is directed to account page
8. Instructor clicks "edit account" link
9. Instructor is directed to account editing page
10. Instructor edits account
11. Instructor clicks "save changes"
12. Instructor is directed to account page
13. Instructor data is updated in database
14. Use case ends
Use Cases: Delete Teacher

DELETE TEACHER
This use case describes how an admin would delete a teacher account

<table>
<thead>
<tr>
<th>USE CASE: DELETE TEACHER</th>
<th>ID: 4C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
<td>Primary Actor: Admin</td>
</tr>
<tr>
<td>Preconditions: None</td>
<td>Postconditions: Teacher account is deleted</td>
</tr>
</tbody>
</table>

Main Flow:
1. Admin goes to www.lolaartswi.com
2. Admin clicks "Administration" link at bottom of home page
3. Admin is directed to admin log in page
4. Admin enters username in username textbox
5. Admin enters password in password textbox
6. Admin clicks "log in" link
7. Admin is directed to Administration page
8. Admin clicks "delete instructor account"
9. Admin is directed to account deletion page
10. Admin selects instructor account to delete
11. Message box "Are you sure you want to delete this account?"
12. Admin clicks "yes"
13. Admin is directed to successful account deletion page "The account has been successfully deleted"
14. Admin account in removed from database
15. Use case ends
Use Cases: Add Course

ADD COURSE

This use case describes how an admin would create a course

<table>
<thead>
<tr>
<th>USE CASE: ADD COURSE</th>
<th>ID: 4D</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Risk Level:</strong> Low</td>
<td></td>
</tr>
<tr>
<td><strong>Primary Actor:</strong> Admin</td>
<td></td>
</tr>
<tr>
<td><strong>Preconditions:</strong> None</td>
<td><strong>Postconditions:</strong> Course is created</td>
</tr>
</tbody>
</table>

**Main Flow:**
1. Admin goes to www.lolaartswi.com
2. Admin clicks on "administration" link at bottom of home page
3. Admin is directed to admin log in page
4. Admin enters username in username text box
5. Admin enters password in password text box
6. Admin clicks "log in" link
7. Admin is directed to admin account page
8. Admin clicks "add class" link
9. Admin is directed to class creation page
10. Admin enters class title in title textbox
11. Admin enters class instructor in instructor textbox
12. Admin enters class description in description textbox
13. Admin selects days class meets from days options
14. Admin selects time class meets from time options
15. Admin selects start date from calendar widget
16. Admin selects end date from calendar widget
17. Admin uploads photos into optional photos
18. Admin enters class cost in cost textbox
19. Admin enters max enrolment in max enrolment textbox
20. Admin clicks "add class"
21. Admin is directed to new page "Class Name' has been created"
22. Course is added to class available page
23. Course data is added to database
24. Admin receives confirmation email
25. Instructor assigned to class receives confirmation email
26. Use case ends
The LOLA administrator selects "add class"

The system creates a new class

addClass( "UML" )

<<Create>>

UML: Class

:ClassManagementSystem

:LOLA Admin
Use Cases: Modify Course

MODIFY COURSE
This use case describes how an instructor would modify a course

USE CASE: MODIFY COURSE

<table>
<thead>
<tr>
<th>ID: 4D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
</tr>
<tr>
<td>Preconditions: None</td>
</tr>
</tbody>
</table>

Main Flow:
1. Instructor goes to www.lolaartswi.com
2. Instructor clicks "instructors" link at bottom of home page
3. Instructor is directed to instructor log in page
4. Instructor enters username in username text box
5. Instructor enters password in password textbox
6. Instructor clicks "log in"
7. Instructor is directed to account page
8. Instructor clicks desired class from "My classes" box
9. Instructor is directed to class page
10. Instructor clicks "edit class" link
11. Instructor is directed to edit class info page
12. Instructor edits class info
13. Instructor clicks "save changes" at bottom of page
14. Changes are made in database to class
15. Instructor is directed to class info page
16. Use case ends
Use Cases: Delete Course

DELETE COURSE
This use case describes how an admin would delete a course

USE CASE: DELETE COURSE
ID: 4D
Risk Level: Low
Primary Actor: Admin
Preconditions: None
Postconditions: Course is deleted

Main Flow:
1. Admin goes to www.lolaartswi.com
2. Admin clicks on "administration" link at bottom of home page
3. Admin is directed to admin log in page
4. Admin enters username in username text box
5. Admin enters password in password text box
6. Admin clicks "log in" link
7. Admin is directed to admin account page
8. Admin clicks "delete class" link
9. Admin is directed to class deletion page
10. Admin clicks desired class
11. Message box "Are you sure you want to delete this class?"
12. Admin clicks "yes"
13. Admin is directed to new page "'class' is successfully deleted"
14. Course is removed from database
15. Admin receives confirmation email
16. Use case ends
Use Cases: Log In Teacher

LOG IN
This use case describes how a teacher would log into his or her account.

USE CASE: LOG IN
ID: 4E
Risk Level: Low
Primary Actor: Teacher
Preconditions: Teacher is not logged in
Postconditions: Teacher is logged in

Main Flow:
1. Instructor goes to www.lolaartswi.com
2. Instructor clicks "instructors" link at bottom of home page
3. Instructor is directed to instructor log in page
4. Instructor enters username in username text box
5. Instructor enters password in password textbox
6. Instructor clicks "log in"
7. Instructor is directed to account page
8. Use case ends
# Use Cases: Add Photo

**ADD PHOTO**

This use case describes how an admin would upload photos

## USE CASE: ADD PHOTO

<table>
<thead>
<tr>
<th>Risk Level: Low</th>
<th>Primary Actor: Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions: None</td>
<td>Postconditions: Photo is uploaded</td>
</tr>
</tbody>
</table>

### Main Flow:
1. Admin goes to www.lolaartswi.com
2. Admin clicks on "administration" link at bottom of home page
3. Admin is directed to admin log in page
4. Admin enters username in username text box
5. Admin enters password in password text box
6. Admin clicks "log in" link
7. Admin is directed to admin account page
8. Admin clicks "photo gallery" link
9. Admin is directed to photo gallery page
10. Admin clicks "upload photos"
11. File upload box appears
12. Admin selects desired photos
13. Admin clicks "upload" button
14. File upload box disappears
15. Uploaded photos appear in gallery
16. Use case ends

![Diagram of aAdmin and :addPhoto](image-url)
Use Cases: Modify Photo

MODIFY PHOTO
This use case describes how an admin would modify a photo

<table>
<thead>
<tr>
<th>USE CASE: MODIFY PHOTO</th>
<th>ID: 5C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
<td></td>
</tr>
<tr>
<td>Primary Actor: Admin</td>
<td></td>
</tr>
<tr>
<td>Preconditions: None</td>
<td>Postconditions: Photo is modified</td>
</tr>
</tbody>
</table>

Main Flow:
1. Admin goes to www.lolaartswi.com
2. Admin clicks on "administration" link at bottom of home page
3. Admin is directed to admin log in page
4. Admin enters username in username text box
5. Admin enters password in password text box
6. Admin clicks "log in" link
7. Admin is directed to admin account page
8. Admin clicks "photo gallery" link
9. Admin is directed to photo gallery page
10. Admin clicks "edit photos"
11. Admin directed to photo gallery editing page
12. Admin edits photos (rearranging, adding comments)
13. Admin clicks "save changes"
14. Admin is directed to photo gallery page
15. Use case ends
Use Cases: Delete Photo

DELETE PHOTO
This use case describes how an admin would delete photos

**USE CASE: DELETE PHOTO**

<table>
<thead>
<tr>
<th>Risk Level: Low</th>
<th>Postconditions: Photo is deleted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Actor: Admin</td>
<td></td>
</tr>
<tr>
<td>Preconditions: None</td>
<td></td>
</tr>
</tbody>
</table>

**Main Flow:**
1. Admin goes to www.lolaartswi.com
2. Admin clicks on "administration" link at bottom of home page
3. Admin is directed to admin log in page
4. Admin enters username in username text box
5. Admin enters password in password text box
6. Admin clicks "log in" link
7. Admin is directed to admin account page
8. Admin clicks "photo gallery" link
9. Admin is directed to photo gallery page
10. Admin clicks "delete photos" link
11. Admin selects photos to delete
12. Admin clicks "delete" link
13. Message box "Are you sure you want to delete the selected photos?"
14. Admin clicks ok
15. Photos are deleted
16. Admin is directed to photo gallery page
17. Use case ends
Use Cases: Add Event

ADD EVENT
This use case describes how an admin would add an event

<table>
<thead>
<tr>
<th>USE CASE:ADD EVENT</th>
<th>ID: 5E</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Risk Level:</strong> Low</td>
<td><strong>Primary Actor:</strong> Admin</td>
</tr>
<tr>
<td><strong>Preconditions:</strong> None</td>
<td><strong>Postconditions:</strong> Event is created</td>
</tr>
</tbody>
</table>

**Main Flow:**
1. Admin goes to www.lolaartswi.com
2. Admin clicks on "administration" link at bottom of home page
3. Admin is directed to admin log in page
4. Admin enters username in username text box
5. Admin enters password in password text box
6. Admin clicks "log in" link
7. Admin is directed to admin account page
8. Admin clicks on "Add event" link
9. Admin is directed to add event page
10. Admin enters selects event date from calendar widget
11. Admin enters event start time in start time textbox
12. Admin enters event end time in end time textbox
13. Admin enters location in location textbox
14. Admin enter event name in event name textbox
15. Admin enters event info in info textbox
16. Admin enters event price in price textbox
17. Admin enters seats available in seats available textbox
18. Admin clicks "Add to calendar"
19. Admin is directed to administration page
20. Use case ends
Use Cases: Modify Event

MODIFY EVENT
This use case describes how an admin would modify an event

USE CASE: MODIFY EVENT
ID: 5F

<table>
<thead>
<tr>
<th>Risk Level: Low</th>
<th>Primary Actor: Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions: None</td>
<td>Postconditions: Event is modified</td>
</tr>
</tbody>
</table>

Main Flow:
1. Admin goes to www.lolaartswi.com
2. Admin clicks on "administration" link at bottom of home page
3. Admin is directed to admin log in page
4. Admin enters username in username text box
5. Admin enters password in password text box
6. Admin clicks "log in" link
7. Admin is directed to admin account page
8. Admin clicks "Events" link
9. Admin is directed to Events page
10. Admin clicks on desired event
11. Admin is directed to Event info page
12. Admin clicks "Edit" link
13. Page becomes editable
14. Admin makes modifications to event
15. Admin clicks "Save Changes" link at bottom of page
16. Event info in calendar is updated
17. Use case ends
Use Cases: Delete Event

**DELETE EVENT**
This use case describes how an admin would delete an event

<table>
<thead>
<tr>
<th>USE CASE: DELETE EVENT</th>
<th>ID: 5G</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Risk Level:</strong> Low</td>
<td><strong>Primary Actor:</strong> Admin</td>
</tr>
<tr>
<td><strong>Preconditions:</strong> None</td>
<td><strong>Postconditions:</strong> Event is deleted</td>
</tr>
</tbody>
</table>

**Main Flow:**
1. Admin goes to www.lolaartswi.com
2. Admin clicks on "administration" link at bottom of home page
3. Admin is directed to admin log in page
4. Admin enters username in username text box
5. Admin enters password in password text box
6. Admin clicks "log in" link
7. Admin is directed to admin account page
8. Admin clicks "Events" link
9. Admin is directed to Events page
10. Admin clicks on desired event
11. Admin is directed to Event info page
12. Admin clicks "Edit" link
13. Page becomes editable
14. Admin clicks "Delete Event"
15. Message box "Are you sure you want to delete this event?"
16. Admin clicks "Yes"
17. Event is removed from calendar
18. Use case ends
Use Cases: View Data

VIEW DATA
This use case describes how an admin would view collected data

USE CASE: VIEW DATA
ID: 5H
Risk Level: Low
Primary Actor: Admin
Preconditions: None
Postconditions: Admin is directed to data collection page

Main Flow:
1. Admin goes to www.lolaartswi.com
2. Admin clicks "Administration" link at bottom of home page
3. Admin is directed to Administration log in page
4. Admin enters username in username text box
5. Admin enters password in password textbox
6. Admin clicks "log in"
7. Admin is directed to Administration page
8. Admin clicks "view collected data" link
9. Admin directed to data collection page.
10. Use case ends
Use Cases: Create Grant Report

CREATE GRANT REPORT
This use case describes how an admin would create a grant report

USE CASE: CREATE GRANT REPORT
ID: 51
Risk Level: Low
Primary Actor: Admin
Preconditions: None
Postconditions: Grant report is created

Main Flow:
1. Admin goes to www.lolaartswi.com
2. Admin clicks "Administration" link at bottom of home page
3. Admin is directed to Administration log in page
4. Admin enters username in username text box
5. Admin enters password in password textbox
6. Admin clicks "log in"
7. Admin is directed to Administration page
8. Admin clicks "view collected data" link
9. Admin directed to data collection page.
10. Admin clicks “Export Data”
11. Admin selects save location.
12. Admin clicks Export
13. use case ends
Use Cases: Log In Administrator

LOG IN
This use case describes how an admin would log into his or her account

<table>
<thead>
<tr>
<th>USE CASE: LOG IN</th>
<th>ID: 5J</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
<td>Primary Actor: Admin</td>
</tr>
<tr>
<td>Preconditions: Admin is not logged in</td>
<td>Postconditions: Admin is logged in</td>
</tr>
</tbody>
</table>

Main Flow:
1. Admin goes to www.lolaartswi.com
2. Admin clicks "Administration" link at bottom of home page
3. Admin is directed to Administration log in page
4. Admin enters username in username text box
5. Admin enters password in password textbox
6. Admin clicks "log in"
7. Admin is directed to Administration page
8. Use case ends
Use Cases: View Events

VIEW EVENTS
This use case describes how a visitor would view events

<table>
<thead>
<tr>
<th>USE CASE: VIEW EVENTS</th>
<th>ID: 6A</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Risk Level:</strong> Low</td>
<td></td>
</tr>
<tr>
<td><strong>Primary Actor:</strong> Visitor</td>
<td></td>
</tr>
<tr>
<td><strong>Preconditions:</strong> None</td>
<td><strong>Postconditions:</strong> Visitor is directed to events page</td>
</tr>
</tbody>
</table>

**Main Flow:**
1. Visitor goes to www.lolaartswi.com
2. Visitor clicks "events" tab on home page
3. Visitor is directed to events page
4. Use case ends
**Use Cases: View Event**

**VIEW EVENT**
This use case describes how a visitor would view event information

<table>
<thead>
<tr>
<th>USE CASE: VIEW EVENT</th>
<th>ID: 6B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: Low</td>
<td></td>
</tr>
<tr>
<td>Preconditions: None</td>
<td></td>
</tr>
<tr>
<td>Postconditions: Visitor is directed to event information page</td>
<td></td>
</tr>
<tr>
<td>Primary Actor: Visitor</td>
<td></td>
</tr>
</tbody>
</table>

**Main Flow:**
1. Visitor goes to www.lolaartswi.com
2. Visitor clicks "events" tab on home page
3. Visitor is directed to events page
4. Visitor clicks desired event link
5. Visitor is directed to event page
6. Use case ends
Use Cases: Purchase Ticket

PURCHASE TICKET
This use case describes how a visitor would purchase a ticket for an event

USE CASE: PURCHASE TICKET

<table>
<thead>
<tr>
<th>ID: 6C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level: High</td>
</tr>
<tr>
<td>Primary Actor: Visitor</td>
</tr>
<tr>
<td>Preconditions: None</td>
</tr>
<tr>
<td>Postconditions: Ticket is purchased</td>
</tr>
</tbody>
</table>

Main Flow:
1. Visitor goes to www.lolaartswi.com
2. Visitor clicks "events" tab on home page
3. Visitor is directed to events page
4. Visitor clicks desired event link
5. Visitor is directed to event page
6. Visitor enters first name in first name textbox (stripe)
7. Visitor enters last name in last name textbox (stripe)
8. Visitor enters card number in card number textbox (stripe)
9. Visitor enters expiration date in expiration date box (stripe)
10. Visitor enters CVC in CVC textbox (stripe)
11. Visitor enters email in email textbox (stripe)
12. Visitor clicks "purchase ticket" (stripe)
13. Message box "thank you for your purchase"
14. Visitor receives confirmation email
15. Event data in database is updated (seats available -1)
16. Event data on event page is updated (seats available -1)
17. Use case ends
Class Diagram

HOW TO READ THE USE CLASS DIAGRAM

The class diagram displays how each user and action interacts with each other on a large scale.
Data Dictionary

HOW TO READ THE USE DATA DICTIONARY

The data dictionary displays all the different forms of data that the website will generate and used to function. Each data entry has a description. Each entry also holds sub entries containing details about itself and the size and type of data that each sub entry is.

USER

All the different types of users of the website, including students, donors, volunteers, teachers, and administrators will all be generated from an abstract user class which has basic attributes that are shared among all the different user classes.

SESSION

The two different types of sessions, courses and events, will be generated from an abstract session class. This session class has the common attributes that the course and event classes share.
## Student

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Student</strong></td>
<td>A user who is enrolling in courses</td>
<td>studentID</td>
<td>Unique identification number</td>
<td>integer</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>firstName</td>
<td>First name of student</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lastName</td>
<td>Last name of student</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>email</td>
<td>Email address of student (also serves as login)</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>password</td>
<td>Password for account login</td>
<td>string</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>address</td>
<td>Address of student</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>city</td>
<td>City of student</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>state</td>
<td>State of student</td>
<td>string</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>zip</td>
<td>Zip code of student</td>
<td>integer</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>phone</td>
<td>Phone number of student</td>
<td>string</td>
<td>12</td>
</tr>
</tbody>
</table>
Donor

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Donor</td>
<td>A user is donating to LOLA</td>
<td>donorID</td>
<td>Unique identification number</td>
<td>integer</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>firstName</td>
<td>First name of donor</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lastName</td>
<td>Last name of donor</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>email</td>
<td>Email address of donor (also serves as login)</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>password</td>
<td>Password for account login</td>
<td>string</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>address</td>
<td>Address of donor</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>city</td>
<td>City of donor</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>state</td>
<td>State of donor</td>
<td>string</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>zip</td>
<td>Zip code of donor</td>
<td>integer</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>phone</td>
<td>Phone number of donor</td>
<td>string</td>
<td>12</td>
</tr>
</tbody>
</table>
### Volunteer

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Volunteer</strong></td>
<td>A user is volunteering for LOLA</td>
<td>volunteerID</td>
<td>Unique identification number</td>
<td>integer</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>firstName</td>
<td>First name of volunteer</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lastName</td>
<td>Last name of volunteer</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>email</td>
<td>Email address of volunteer (also serves as login)</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>password</td>
<td>Password for account login</td>
<td>string</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>address</td>
<td>Address of volunteer</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>city</td>
<td>City of volunteer</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>state</td>
<td>State of volunteer</td>
<td>string</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>zip</td>
<td>Zip code of volunteer</td>
<td>integer</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>phone</td>
<td>Phone number of volunteer</td>
<td>string</td>
<td>12</td>
</tr>
</tbody>
</table>
Teacher

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teacher</td>
<td>A user who is teaching courses at LOLA</td>
<td>teacherID</td>
<td>Unique identification number</td>
<td>integer</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>firstName</td>
<td>First name of teacher</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lastName</td>
<td>Last name of teacher</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>email</td>
<td>Email address of teacher (also serves as login)</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>password</td>
<td>Password for account login</td>
<td>string</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>address</td>
<td>Address of teacher</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>city</td>
<td>City of teacher</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>state</td>
<td>State of teacher</td>
<td>string</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>zip</td>
<td>Zip code of teacher</td>
<td>integer</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>phone</td>
<td>Phone number of teacher</td>
<td>string</td>
<td>12</td>
</tr>
</tbody>
</table>
## Administrator

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin</td>
<td>Person with administrator account and privileges</td>
<td>adminID</td>
<td>Unique identification number</td>
<td>integer</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>firstName</td>
<td>First name of administrator</td>
<td>string</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lastName</td>
<td>Last name of administrator</td>
<td>string</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>email</td>
<td>Email/username of administrator</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>password</td>
<td>Password for administrator account</td>
<td>string</td>
<td>15</td>
</tr>
</tbody>
</table>
User

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Abstract class containing all user classes</td>
<td>userID</td>
<td>Unique identification number</td>
<td>integer</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>firstName</td>
<td>First name of user</td>
<td>string</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lastName</td>
<td>Last name of user</td>
<td>string</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>email</td>
<td>Email address/username of user</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>password</td>
<td>Password related to user account</td>
<td>string</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>userType</td>
<td>Type of user account (admin, student, teacher, donor, volunteer)</td>
<td>string</td>
<td>20</td>
</tr>
</tbody>
</table>
## Session

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session</td>
<td>Abstract class for an event or course LOLA is hosting</td>
<td>sessionID</td>
<td>Unique identification number</td>
<td>integer</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sessionName</td>
<td>Name of the event</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sessionDescription</td>
<td>Description of the event</td>
<td>string</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sessionSDate</td>
<td>Date of the event</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sessionEDate</td>
<td>End date of session</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sessionSTime</td>
<td>Start time of session</td>
<td>string</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sessionETime</td>
<td>End time of session</td>
<td>string</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sessionPrice</td>
<td>Price of session</td>
<td>Decimal</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sessionSeats</td>
<td>Seats available for session</td>
<td>Int</td>
<td>3</td>
</tr>
</tbody>
</table>
## Course

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course</td>
<td>A course offered to students at LOLA</td>
<td>courseID</td>
<td>Unique identification number</td>
<td>integer</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>courseName</td>
<td>Name of the course</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>courseDescription</td>
<td>Description of the course</td>
<td>string</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>courseSDate</td>
<td>Start date of the course</td>
<td>string</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>courseEDate</td>
<td>End date of the course</td>
<td>string</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>courseSTime</td>
<td>Start time of the course</td>
<td>string</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>courseETime</td>
<td>End time of the course</td>
<td>string</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>coursePrice</td>
<td>Price of enrollment of course</td>
<td>integer</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>courseSeats</td>
<td>Number of seats available in course</td>
<td>Integer</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>teacherID</td>
<td>Teacher ID of related teacher</td>
<td>Integer</td>
<td>10</td>
</tr>
</tbody>
</table>
# Event

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>An event which LOLA hosts and sells tickets for</td>
<td>eventName</td>
<td>Name of the event</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eventDescription</td>
<td>Description of the event</td>
<td>string</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eventDate</td>
<td>Date of the event</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eventSTime</td>
<td>Start time of the event</td>
<td>string</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eventETime</td>
<td>End time of the event</td>
<td>string</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eventPrice</td>
<td>Price of tickets for the event</td>
<td>string</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eventSeats</td>
<td>Number of seats available for event</td>
<td>integer</td>
<td>3</td>
</tr>
</tbody>
</table>
## Donation

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Donation</td>
<td>A monetary donation made by a donor with account</td>
<td>donationID</td>
<td>Unique identification number</td>
<td>integer</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>donationAmount</td>
<td>Amount of donation</td>
<td>Decimal</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>donationStatus</td>
<td>Status of donation (either one-time or recurring)</td>
<td>int</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>donationSDate</td>
<td>Start date of recurring donation</td>
<td>string</td>
<td>10</td>
</tr>
<tr>
<td>Duration</td>
<td></td>
<td></td>
<td>Duration of recurring donation</td>
<td>Int</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>donationRecursion</td>
<td>Number of times the donation will reoccur in desired period</td>
<td>Int</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>donationRecursionType</td>
<td>Period of donation recursion (monthly, quarterly, semi-annually, annually)</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>donorID</td>
<td>Donor ID of related donor</td>
<td>Int</td>
<td>10</td>
</tr>
</tbody>
</table>
# Photo

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Photo</td>
<td>A photo uploaded to LOLA's website</td>
<td>photoID</td>
<td>Unique identification number</td>
<td>integer</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>photoName</td>
<td>Name of the photo</td>
<td>string</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>photoDescription</td>
<td>Description of the photo</td>
<td>string</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sessionId</td>
<td>Session ID of related course or event</td>
<td>Int</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>addedBy</td>
<td>Name of admin who uploaded photo</td>
<td>string</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dateAdded</td>
<td>Date when photo was uploaded</td>
<td>string</td>
<td>10</td>
</tr>
</tbody>
</table>
Grant Report

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grant Report</td>
<td>A grant report generated by LOLA admin</td>
<td>donorData</td>
<td>Donor table containing data related to donors and donations</td>
<td>xml</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>studentData</td>
<td>Student table containing data related to students and classes</td>
<td>xml</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>eventData</td>
<td>Event table containing data related to events and ticket sales</td>
<td>xml</td>
<td></td>
</tr>
</tbody>
</table>
## Survey

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survey</td>
<td>A survey completed by LOLA user</td>
<td>surveyID</td>
<td>Unique identification number</td>
<td>integer</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>userID</td>
<td>ID of user completing survey</td>
<td>Int</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>survey</td>
<td>Survey created by user</td>
<td>string</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sessionID</td>
<td>ID of session surveyed</td>
<td>Int</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>date</td>
<td>Date of survey</td>
<td>string</td>
<td>15</td>
</tr>
</tbody>
</table>
## Stripe Payment

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Entity Description</th>
<th>Column Name</th>
<th>Column Description</th>
<th>Data Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stripe Payment</td>
<td>Payment made to LOLA using Stripe Payment System</td>
<td>Email</td>
<td>Email address of user making payment</td>
<td>String</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cardNumber</td>
<td>Card number of card making payment</td>
<td>Int</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>expDate</td>
<td>Expiration date of related card</td>
<td>Int</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cvc</td>
<td>CVC code of related card</td>
<td>Int</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>paymentType</td>
<td>Type of payment (donation, tuition, ticket purchase)</td>
<td>string</td>
<td>20</td>
</tr>
</tbody>
</table>
Windows Navigation Diagrams

A Windows Navigation Diagram shows how the system is laid out starting with the landing page, which in this case would be the home page of the site. It displays the buttons or links that are needed to be clicked to navigate to the next screen. The lines show how one screen are connected to another. The main links from the home page are Classes/Events landing page, Donation Landing Page, and the LOLA Lovers Landing page. The LOLA Lovers Landing page is where teachers, volunteers and Admins will go to log in to get to more functionality.
Physical Architecture Design
Design Procedures for Security

OUR SECURITY
As mentioned in the nonfunctional requirements, the staff will have access to the system through any web browser. The user will have to authenticate an employee username and password every time they log into the system regardless of browser.

Students
Students will need to provide both username and password for verification when logging into their account.

Donors
Donors will need to provide both username and password for verification when logging into their account.

Teachers
Teachers will need to provide both username and password for verification when logging into their account.

Volunteers
Volunteers will need to provide both username and password for verification when logging into their account.

Administrators
Administrators will need to provide both username and password for verification when logging into their account.

A SMALL ORANGE SECURITY
The online database and backup software we plan to incorporate proves 24/7/365 monitoring and provides alerts for any network update that could affect your service
Gantt Chart

0. Use Case Description - Seshan Gardner
1. Use Case Diagram - Bradley Erickson
2. Use Case HTML Prototype - Spencer Kerber
3. Class Diagram - Spencer Kerber
4. Database Design - Deshan Gardner
5. Data Definitions - Spencer Kerber
6. UI and windows navigation diagram - Christopher James
7. Prototypes - Christopher James
8. Physical Architecture Design - Spencer Kerber
9. Design Procedures for security concerns and non-functional requirements - Deshan Gardner
10. Elaboration Phase prototype - Christopher James
11. Gantt Chart - Bradley Erickson
Prototypes
Main Page
View Classes/Events Page
View Class/Event

Diagram of a website interface showing a page for viewing a class or event. The page includes sections for donating, LOLA logo, and information about the selected class. There is also a button for signing up for the class. A note indicates that when a teacher creates a class, they will be able to upload pictures they want to display when a user looks at the class. They can be from previous classes or whatever the teacher wants to upload.
Register for Class/Event page
Donate page

![Diagram of a donation page for LOLA Artisans]

- Some text on why donating to LOLA is a good idea. What it means to their community and their members.

- **LOLA Logo**

- **First Name**
  - **Last Name**

- **Address Line One**
  - **Address Line Two**

- **Country**
  - **State**

- **Zip Code**
  - **Phone Number**

- **What Project Do You Want To Donate Money Towards?**
  - Allow users to select projects that LOLA wants money to be donated towards. Could be new classroom, buy equipment, etc.

- **Payment Info**
  - **Name On Card**
    - **Card Number**
  - **Amount To Donate**
    - **Number On Back Of Card**
    - **Expiration Date**

- **Process Donation**
Donation Successful Page

Thank you for donating! Land O’ Lakes Artisans

Thank you so much for donating to LOLA! Your money will go towards ((Project selected on previous page )) which will allow LOLA to be able to better support our community and our members. We greatly appreciate your support.
LOLA Lovers Landing Page

Welcome LOLA Lovers!

Not yet a LOLA Lovers? Interested in helping us grow and do more for our community?
Click the button below to sign up to be a LOLA Lover! We greatly appreciate all of our volunteers!

Sign Up Now!

Upcoming Events

FEB 2008

1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29
Volunteer Login Page
Volunteer View Event Page
Volunteer Registration Page
Modify Volunteer Account Page
Admin Login Page

You must have admin access to reach this part of the site. Please contact the owner to get this access.

Email
Password
Login

Forgot Password? Click The Button Below
Forgot Password
Admin Dashboard Page

Welcome LOLA Admin!

- View All Accounts
- Create Admin Account
- Create Event/Class
- View All Events
- Create Teacher Account
- View Report Page
Grant Report Page

Chart Showing Total Students Per Month

More Reports

A drop-down that has a list of other quick reports (Money spent, Donations, etc)

Other Tables

Data Range

Ability to select to view other database tables and date ranges.

Export To CSV

Grant Report - Land O' Lakes Artisans

Brief Summary of LOLA and how they impact their community.

LOLA Logo

Name Of Student | Age | Address
---|---|---
Giacomo Guliemeni | 8 | Address
Marco Betton | 12 | Address
Marina Mocchion | 6 | Address
Valerie Liberty | 14 | Address
Guido Jock Guliemeni | 15 | Address

http://www.lolartassociates.com/admin/grant-report

Search

Become a Member Today!

Facebook | Twitter | Instagram
Create Admin Account Page

[Image of the Create Admin Account Page]

Create A New Admin User. Please fill out the information below.
An Email will be sent to the user to set their password.

First Name: [Field]
Last Name: [Field]
Email: [Field]

Send Account Creation Email
View All Accounts Page
View All Classes/Events page
Add Volunteer Page
Add Teacher page

[Diagram of Teacher Account Creation page]

- Enter First Name
- Enter Last Name
- Enter Email Address
- Enter Password
- Re-Enter Password

- Enter Address
- Enter City
- Select State
- Enter Zip
- Enter Phone Number

Create Teacher Account
Modify Admin Account Page
Create Event Page
Edit Event Page
Login Teacher Page

![Login Teacher Page Diagram]
Teacher Landing Page
Modify Teacher Page

[Image of a web page interface for modifying teacher accounts]
Purchase Event Ticket Page