table of contents

Presented by: tristan weihmann

Shane Ritz

Lynnzee Matthews

Matthew Floyd

Dylan Claywell



Relay Engine

Version [1.0]

[Executive summary 2](#_Toc437536262)

[Summary of Proposal 2](#_Toc437536263)

[Deliverables 2](#_Toc437536264)

[seperate Documents/ files 3](#_Toc437536265)

[Relay Engine Tasks 4](#_Toc437536266)

[Lynnzee 4](#_Toc437536267)

[Matthew 4](#_Toc437536268)

[Shane 4](#_Toc437536269)

[Dylan 4](#_Toc437536270)

[Tristan 4](#_Toc437536271)

[git 5](#_Toc437536272)

[link to repository 5](#_Toc437536273)

[What is git? 5](#_Toc437536274)

[Website Hierarchy 6](#_Toc437536275)

[database entity relationship diagram 7](#_Toc437536276)

[DATABASE DATA DICTIONARY 8](#_Toc437536277)

[DATABSE SQL STATEMENTS 9](#_Toc437536278)

[TABLE CREATION 9](#_Toc437536279)

[DATA INSERTION 10](#_Toc437536280)

Executive summary

# Summary of Proposal

Evansville Day School is looking for a web application that will allow them to send text messages to students who sign up for alert lists. We have been given a document outlining the functionality and basic design of the web application. The application requires the use of a database in order to keep track of users, senders, lists, which lists users are subscribed to, and which senders have authorization to send messages to users through the lists. The administrator will have a table in the database which keeps track of and manages how many tokens the senders have for use, the logo image, and the organizations security code.

The Visual Studio project will be hosted on github and will be managed through the git source control method and will be further explained later in the document.

# Deliverables

In this document we will provide a few different deliverables which will highlight the objects designed to complete this web application. The user manual for the application has already been created by the web designers, and will be included in a separate document. The database entity relationship diagram, the website hierarchy, and data dictionary will be included in this document. The SQL statements used to implement the database will be included in this document. The visual studio project will be delivered as a separate attachment, and the github repository link will be provided in this document. The database will be hosted online through Azure. Relay engine tasks is a document that was originally made to serve as a schedule of soft deadlines. It describes who was supposed to be in charge of what page.

**THE ADMIN**

**USERNAME = admin**

**PASSWORD = admin**

**Sender**

**EMAIL =** [**euismod.et.commodo@infelis.com**](mailto:euismod.et.commodo@infelis.com)

**PASSWORD = YOURA**

seperate Documents/ files

* User Manual : RelayEngine.pdf
* Visual Studio project: CS478-RelayEngine.zip

Relay Engine Tasks

Here’s the list of everything we need to do for this project. I’ve divided the work on a per person basis, and each object has their own due dates set.

These are soft deadlines, but try and keep close to them as some elements rely on the existence of other ones. If you can finish a task before the given deadline, do it and send an email to let us know so people can start their tasks earlier.

You can also design some of your web pages early, and just change a few things later so the previous page is directing to it properly.

The person in charge of making each web page is also going to be in charge of filling the tables/lists on that page using the database we create.

# Lynnzee

* Home Page – 20 November
* Alert Registration confirmation page – 2 December
* Manage lists page – 8 December
* Account information page – 8 December

# Matthew

* Admin sign in page – 26 November
* Send Message Page – 2 December
* Add tokens page – 8 December

# Shane

* ERD Design – 19 November
* Text message page – 8 December
* Sent message confirmation page – 8 December

# Dylan

* Text alert registration page – 26 November
* Organization management page – 2 December
* Manage senders page – 8 December
* Sender list management page – 8 December

# Tristan

* Create and fill database – 22 November
* Update logo page – 8 December
* Organization Security code page – 8 December
* User Manual – 8 December
* Manage list page – 8 December

git

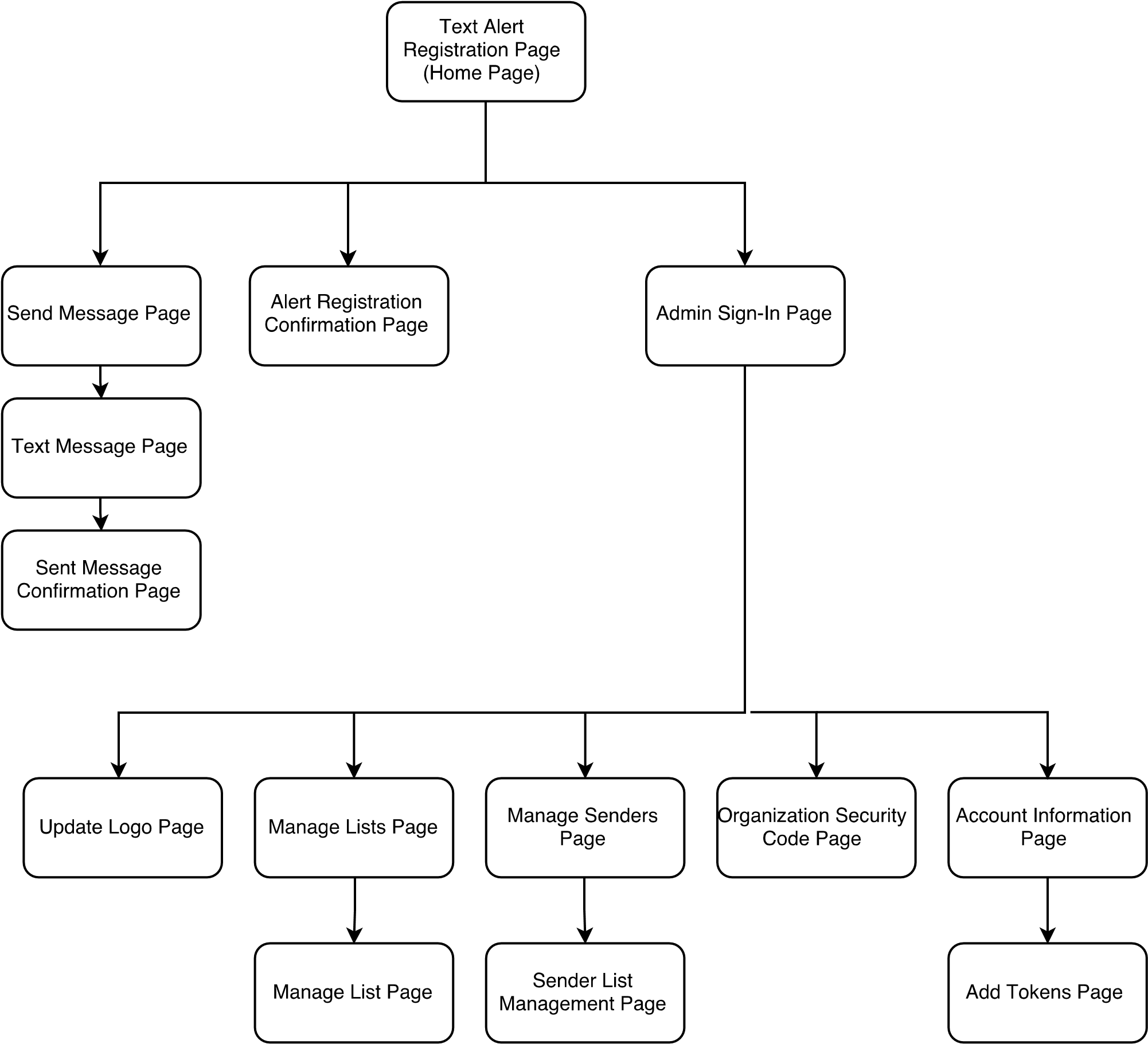
# link to repository

<https://github.com/usiwallaby/CS478-RelayEngine>

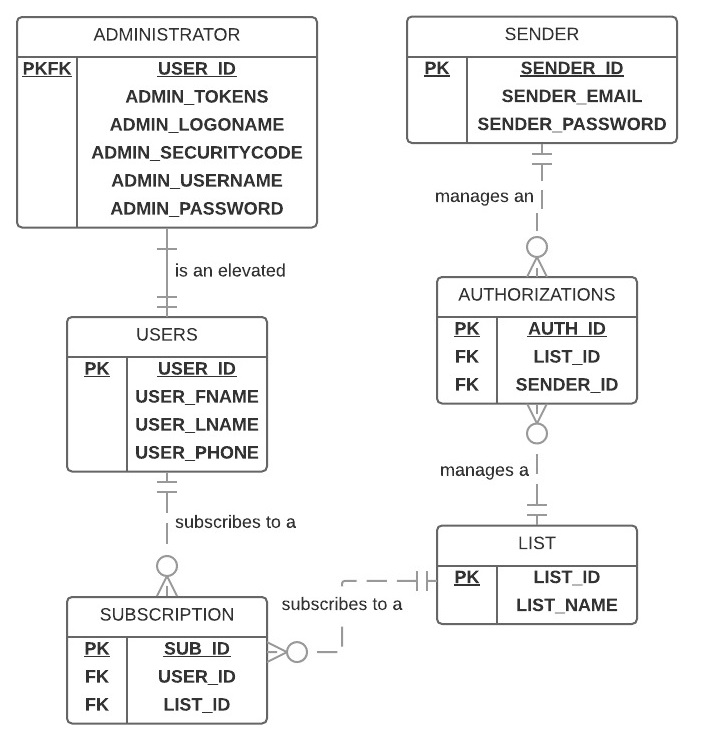
# What is git?

Git is a version control method which allows multiple people to make changes to a project at the same time. It keeps track of what changes are made, when the changes are made, and who made the changes. It also allows you to revert to any version of a project if you need to. Github is a website that simply hosts projects and acts as a hub for people to interact with a project using git.

Website Hierarchy



database entity relationship diagram



DATABASE DATA DICTIONARY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Type/Size** | **Validation** | **Key** |
| ADMINISTRATOR | USER\_ID | INT | REQUIRED | PRIMARY/ FOREIGN |
|  | ADMIN\_TOKENS | INT | REQUIRED |  |
|  | ADMIN\_LOGONAME | VARCHAR(50) | REQUIRED |  |
|  | ADMIN\_SECURITYCODE | VARCHAR(50) | REQUIRED |  |
|  | ADMIN\_USERNAME | VARCHAR(20) | REQUIRED |  |
|  | ADMIN\_PASSWORD | VARCHAR(20) | REQUIRED |  |
| AUTHORIZATIONS | AUTH\_ID | INT | REQUIRED | PRIMARY |
|  | LIST\_ID | INT | REQUIRED | FOREIGN |
|  | SENDER\_ID | INT | REQUIRED | FOREIGN |
| LIST | LIST\_ID | INT | REQUIRED | PRIMARY |
|  | LIST\_NAME | VARCHAR(50) | REQUIRED |  |
| SENDER | SENDER\_ID | INT | REQUIRED | PRIMARY |
|  | SENDER\_EMAIL | VARCHAR(30) | REQUIRED |  |
|  | SENDER\_PASSWORD | VARCHAR(20) | REQUIRED |  |
| SUBSCRIPTION | SUB\_ID | INT | REQUIRED | PRIMARY |
|  | USER\_ID | INT | REQUIRED | FOREIGN |
|  | LIST\_ID | INT | REQUIRED | FOREIGN |
| USERS | USER\_ID | INT | REQUIRED | PRIMARY |
|  | USER\_FNAME | VARCHAR(20) | REQUIRED |  |
|  | USER\_LNAME | VARCHAR(20) | REQUIRED |  |
|  | USER\_PHONE | CHAR(10) | REQUIRED |  |

DATABSE SQL STATEMENTS

# TABLE CREATION

CREATE TABLE USERS(

USER\_ID INT UNIQUE NOT NULL,

USER\_FNAME VARCHAR(20) NOT NULL,

USER\_LNAME VARCHAR(20) NOT NULL,

USER\_PHONE CHAR(10) NOT NULL,

PRIMARY KEY (USER\_ID)

);

CREATE TABLE ADMINISTRATOR(

USER\_ID INT UNIQUE NOT NULL,

ADMIN\_TOKENS INT NOT NULL,

ADMIN\_LOGONAME VARCHAR(50) NOT NULL,

ADMIN\_SECURITYCODE VARCHAR(50) NOT NULL,

PRIMARY KEY (USER\_ID),

FOREIGN KEY (USER\_ID) REFERENCES USERS(USER\_ID)

);

CREATE TABLE LIST(

LIST\_ID INT UNIQUE NOT NULL,

LIST\_NAME VARCHAR(50) NOT NULL,

PRIMARY KEY (LIST\_ID)

);

CREATE TABLE SUBSCRIPTION(

SUB\_ID INT UNIQUE NOT NULL,

USER\_ID INT NOT NULL,

LIST\_ID INT NOT NULL,

PRIMARY KEY (SUB\_ID),

FOREIGN KEY (USER\_ID) REFERENCES USERS(USER\_ID),

FOREIGN KEY (LIST\_ID) REFERENCES LIST(LIST\_ID)

);

CREATE TABLE SENDER(

SENDER\_ID INT UNIQUE NOT NULL,

SENDER\_EMAIL VARCHAR(30) NOT NULL,

PRIMARY KEY (SENDER\_ID)

);

CREATE TABLE AUTHORIZATIONS(

AUTH\_ID INT UNIQUE NOT NULL,

LIST\_ID INT NOT NULL,

SENDER\_ID INT NOT NULL,

PRIMARY KEY (AUTH\_ID),

FOREIGN KEY (LIST\_ID) REFERENCES LIST(LIST\_ID),

FOREIGN KEY (SENDER\_ID) REFERENCES SENDER(SENDER\_ID)

);

# DATA INSERTION

INSERT INTO USERS(USER\_ID,USER\_FNAME,USER\_LNAME,USER\_PHONE)

VALUES (0,'Lamar','Rodgers','3281763886'),(1,'Tashya','Griffith','3878693363'),(2,'Echo','Sims','7036598094'),(3,'Cassidy','Mitchell','3728051597'),(4,'Signe','Miles','5353534416'),(5,'Fleur','Henderson','8786273576'),(6,'Juliet','Buck','5562156285'),(7,'Melinda','Melton','4239725073'),(8,'Amir','Tucker','7605313408'),(9,'Lareina','Austin','1311013540'),(10,'Clementine','Beasley','5448144124'),(11,'Alice','Figueroa','1979075262'),(12,'Florence','Hardy','9991881621'),(13,'Miranda','Marshall','4355559997'),(14,'Gray','Montoya','9217741224'),(15,'Charlotte','Conley','1757792725'),(16,'Sonya','Wilson','2857899183'),(17,'Mira','Ayers','9991608560'),(18,'Zoe','Strong','4707139747'),(19,'Shea','Osborne','9977149890'),(20,'Chantale','Newton','4516821700'),(21,'Aretha','Cameron','4043331868'),(22,'Ruby','Washington','7509160764'),(23,'Leilani','Jensen','4915654438'),(24,'Ciara','Tucker','6235401497'),(25,'Marcia','Oconnor','4031223907'),(26,'Neve','Key','6093873848'),(27,'Jamalia','Garza','7865590442'),(28,'Todd','Keith','2447028399'),(29,'Tana','Cote','8735517003'),(30,'Conan','Mcdonald','5377350659'),(31,'Kieran','Olson','5808790301'),(32,'Daquan','Guerra','6678879734'),(33,'Evelyn','Singleton','4937834966'),(34,'Elliott','Torres','1677124269'),(35,'Brennan','Graham','4719343621'),(36,'Lillith','Knox','8553662283'),(37,'Maia','Floyd','9636619347'),(38,'Ferris','Holden','3419882142'),(39,'Rigel','Miranda','5971743681'),(40,'Malachi','Faulkner','5164588421'),(41,'Lysandra','Casey','1033034996'),(42,'Xyla','Mcfadden','1702261221'),(43,'Dale','Zimmerman','9298358224'),(44,'Drew','Sanders','9475763890'),(45,'Beverly','Mueller','8936483384'),(46,'Sawyer','Maxwell','4978856158'),(47,'Jillian','Waters','7446029350'),(48,'Sage','Hernandez','9664570556'),(49,'Zeph','Dean','4269191867'),(50,'Barry','Torres','7871059211'),(51,'Whitney','Hood','2095318191'),(52,'Deirdre','Sweet','5966091940'),(53,'Lisandra','Williams','2308050125'),(54,'Orlando','Blevins','8175284733'),(55,'Victoria','Yates','6612132375'),(56,'Portia','Houston','1016319164'),(57,'Petra','Weeks','1076286770'),(58,'Garrett','Freeman','5963925737'),(59,'Meghan','Boyd','4059065995'),(60,'Deborah','Porter','1187213667'),(61,'Leonard','Combs','1418917924'),(62,'Fulton','Noel','1307673549'),(63,'Beatrice','Maxwell','8703776611'),(64,'Deacon','Acevedo','2922112042'),(65,'Mercedes','Ewing','4222743137'),(66,'Raya','Strickland','2848152778'),(67,'Wendy','King','5057335493'),(68,'Donna','Mack','2827489012'),(69,'Laith','Patton','8279138708'),(70,'Jescie','Kelley','6135479892'),(71,'Tiger','Bonner','6546881085'),(72,'Idona','Simpson','4225231902'),(73,'Darrel','Manning','6499675023'),(74,'Aiko','Brennan','6674597016'),(75,'Nadine','Drake','8513947266'),(76,'Indira','Keller','8827375121'),(77,'Salvador','Barrera','3064789405'),(78,'Shay','Mills','8009740504'),(79,'Imani','Black','6749266112'),(80,'Ila','Vazquez','6143501236'),(81,'Graiden','Finch','8751436360'),(82,'Avram','Garza','8362202002'),(83,'Maggie','Crawford','8402112243'),(84,'Demetria','Dejesus','9768326045'),(85,'Libby','Gibson','4378619705'),(86,'Calvin','Donaldson','6076403272'),(87,'Shoshana','Evans','8847418917'),(88,'Ivor','Mccoy','2095363905'),(89,'Germane','Callahan','5341719792'),(90,'Elvis','Sharp','5575492929'),(91,'Athena','Knapp','1076878726'),(92,'Kamal','Marsh','4539473938'),(93,'Steel','Farrell','6553867276'),(94,'Madonna','Klein','1882900527'),(95,'Steven','Levine','6621665468'),(96,'Raymond','Harrington','8447145251'),(97,'Lareina','Gibson','6524984428'),(98,'Michael','Mcgee','4915018818'),(99,'Amy','Stout','9815337379')

;

INSERT INTO LIST(LIST\_ID,LIST\_NAME)

VALUES (0,'School Cancellation/Delays'),(1,'Lower School Sports'),(2,'Upper School Sports'),(3,'Lower School General Announcements'),(4,'Upper School General Announcements')

;

INSERT INTO SUBSCRIPTION(SUB\_ID,USER\_ID,LIST\_ID)

VALUES (0,1,1),(1,2,0),(2,3,3),(3,4,0),(4,5,2),(5,6,4),(6,7,3),(7,8,1),(8,9,2),(9,10,3),(10,11,0),(11,12,3),(12,13,2),(13,14,3),(14,15,2),(15,16,1),(16,17,0),(17,18,4),(18,19,0),(19,20,2)

;

INSERT INTO SENDER(SENDER\_ID,SENDER\_EMAIL)

VALUES (0,'euismod.et.commodo@infelis.com'),(1,'at.arcu@faucibusorciluctus.com'),(2,'egestas@mus.com'),(3,'enim@ornareegestas.net'),(4,'Maecenas.libero.est@metus.org')

;

INSERT INTO AUTHORIZATIONS(AUTH\_ID,LIST\_ID,SENDER\_ID)

VALUES (0,0,0),(1,1,1),(2,2,2),(3,3,3),(4,4,4)

;

INSERT INTO ADMINISTRATOR(USER\_ID,ADMIN\_TOKENS,ADMIN\_LOGONAME,ADMIN\_SECURITYCODE)

VALUES (0,5000,'evansville\_day\_school.jpg','No Code Set')

;