Bring Your Daughter to Clemson

Technical Reference Manual

Dorn, Cornejo, Kennedy 1

App Summary

Target Devices

1. iPhone with iOS 8 or later
2. Xcode 6, Swift 1.2

App Functions

1. List of events (pulled from external database)
2. User specific agenda
3. Map of campus with current location and locations of events.
4. Scavenger hunt
5. Contact info for the BYDTC event staff.

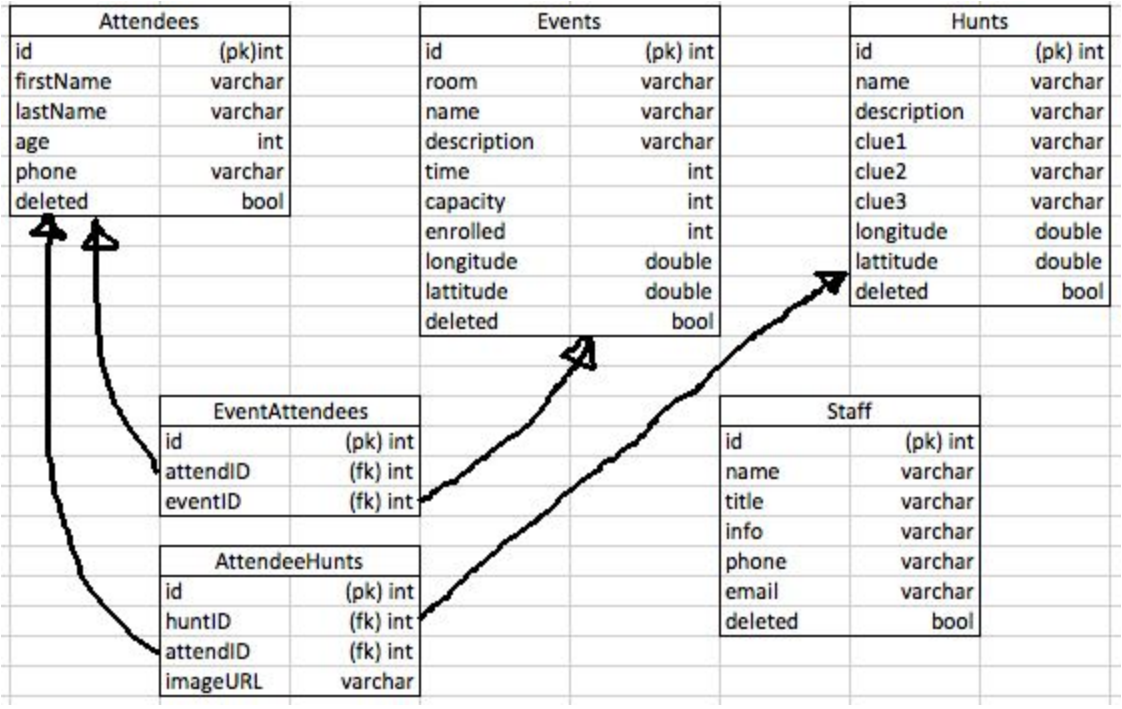
Benefits

- The app will benefit the students and their parents attending the conference. The app will allow for a better organization of the conference and contain features that will enhance overall experience. Since most of the conference has been done on paper in the past, this will allow an easy way for users to participate in the event and will help administrators run the event.

Database Info

External Database

The external database contains all of the the event, attendee, scavenger hunt, and staff member data. See schema below. Each of the 4 main table contain a ‘deleted’ boolean. When this is set to 1, the data will not be downloaded to the internal database. The ‘EventAttendees’ table will contain entries for each event that each attendee has signed up for within the app. The ‘AttendeeHunts’ table will have entries for each scavenger hunt item that each attendee has found. There will not be any artifacts uploaded to the external database.



External Database Schema

Internal Database

The database on the device will contain the events, hunts, and staff tables pulled from the external database . Each of these tables contains all of the information from the external database tables, except for the deleted attribute. When syncing the databases, if the item has ‘deleted’ set to 1, it will be removed from the internal database if it exists, and will otherwise not be loaded into the internal database. The ‘Events’ table has an additional attribute called ‘going’ which will be set to 1 if the current user has signed up to attend that event. The ‘Hunts’ table also has an additional attribute called ‘found’ which will be set to 1 if the current user has found that item.

Events	
id	(pk) int
room	String
name	String
description	String
time	int
capacity	int
enrolled	int
longitude	double
lattitude	double
going	int

Hunts	
id	(pk) int
name	String
description	String
clue	String
longitude	double
lattitude	double
found	int
image	UIImage

Staff	
id	(pk) int
name	varchar
title	varchar
info	varchar
phone	int
email	varchar

Internal Database Schema

Authoring Tool

We use an application (Windows only) called HeidiSQL that is very straightforward to view and edit the external database. Once the database info credentials have been entered, it displays all the data under the data tab for each table listed on the left. The data can be edited, and the database will be updated synchronously. **You must be on Clemson's campus for this to work.**

When starting out with HeidiSQL just click New in the bottom left corner to create a new session. You may name it whatever you want, we just named it BYDTC Data.
The following information must be entered when creating a session in HeidiSQL:

Hostname / IP:	mysql1.cs.clemson.edu
User:	bydtc_bv08
Password:	asdf1234
Databases:	bydtc_0bhv

All other information was left as default.

After this information is entered, click open. You can then click the arrow to the left of the database 'bydtc_0bhv' in the list on the left to expand it. From there you can select any of the tables. Once you have selected a table you can click on the data tab at the top on the right to display all of the contents of that table. Double click on any data item to edit it. You must click away from the selected row for it to update the database.

To download HeidiSQL please visit this link: http://www.heidisql.com/download.php
For a detailed tutorial on HeidiSQL please visit this link: http://www.heidisql.com/help.php

Development team

Evan Dorn
dorn3@g.clemson.edu
Front end view controllers, maps

Chris Cornejo
ccornej@g.clemson.edu
Design, storyboards, documentation

Brad Kennedy
bckenne@g.clemson.edu
Backend databases, authoring tool

Project GitHub

Acknowledgements

Thank you to Ms. Robin Rast, the Clemson Alumni Association, and Dr. Pargas for giving us the opportunity to work on the Bring Your Daughter to Clemson app.

Appendix

getAllEvents.php

this file returns a JSON array containing all information of all Events

```
<?php

$dbhostname = 'mysql1.cs.clemson.edu';

$dbusername = 'bydtc_bv08';

$dbpassword = 'asdf1234';

$conn = mysql_connect($dbhostname, $dbusername, $dbpassword);

if(!$conn)
{

  die('Could not connect: ' . mysql_error());

}
//echo ('connected');

//echo 'MySQL Connected successfully'."<BR>";
mysql_select_db("bydtc_0bhv") or die(mysql_error());

$result = mysql_query("SELECT * FROM Events");

if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
$rows = array();
while($r = mysql_fetch_assoc($result)){
      $rows[] = $r;
}
print json_encode($rows);
mysql_close($conn);
```

getAllHunts.php

this file returns a JSON array containing all information of all Hunts

```
<?php

$dbhostname = 'mysql1.cs.clemson.edu';

$dbusername = 'bydtc_bv08';

$dbpassword = 'asdf1234';

$conn = mysql_connect($dbhostname, $dbusername, $dbpassword);

if(!$conn)
{

  die('Could not connect: ' . mysql_error());

}
//echo ('connected');

//echo 'MySQL Connected successfully'."<BR>";
mysql_select_db("bydtc_0bhv") or die(mysql_error());

$result = mysql_query("SELECT * FROM Hunts");

if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
$rows = array();
while($r = mysql_fetch_assoc($result)){
      $rows[] = $r;
}
print json_encode($rows);
mysql_close($conn);
```

getAllStaff.php

this file returns a JSON array containing all information of all Staff

```
<?php

$dbhostname = 'mysql1.cs.clemson.edu';

$dbusername = 'bydtc_bv08';

$dbpassword = 'asdf1234';

$conn = mysql_connect($dbhostname, $dbusername, $dbpassword);

if(!$conn)
{

  die('Could not connect: ' . mysql_error());

}
//echo ('connected');

//echo 'MySQL Connected successfully'."<BR>";
mysql_select_db("bydtc_0bhv") or die(mysql_error());

$result = mysql_query("SELECT * FROM Staff");

if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
$rows = array();
while($r = mysql_fetch_assoc($result)){
      $rows[] = $r;
}
print json_encode($rows);
mysql_close($conn);
```

getMyEvents.php

this file returns a JSON array containing the eventId of all EventAttendee entries with the userId given as an argument

```
<?php

$dbhostname = 'mysql1.cs.clemson.edu';

$dbusername = 'bydtc_bv08';

$dbpassword = 'asdf1234';

$conn = mysql_connect($dbhostname, $dbusername, $dbpassword);

if(!$conn)
{

  die('Could not connect: ' . mysql_error());

}
//echo ('connected');

if (isset ($_GET["id"]) ){
            $id = $_GET["id"];
}
//echo 'MySQL Connected successfully'."<BR>";
mysql_select_db("bydtc_0bhv") or die(mysql_error());

$result = mysql_query("SELECT eventId FROM EventAttendees WHERE
attendId='$id'");

if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
else{
      $rows = array();
      while($r = mysql_fetch_assoc($result)){
            $rows[] = $r;
      }
      header('Content-Type: application/json');
      print json_encode($rows);
}
mysql_close($conn);
```

getMyHunts.php

this file returns a JSON array containing the huntId of all AttendeeHunts entries with the userId given as an argument

```
<?php

$dbhostname = 'mysql1.cs.clemson.edu';

$dbusername = 'bydtc_bv08';

$dbpassword = 'asdf1234';

$conn = mysql_connect($dbhostname, $dbusername, $dbpassword);

if(!$conn)
{

  die('Could not connect: ' . mysql_error());

}
//echo ('connected');

if (isset ($_GET["id"]) ){
            $id = $_GET["id"];
}
//echo 'MySQL Connected successfully'."<BR>";
mysql_select_db("bydtc_0bhv") or die(mysql_error());

$result = mysql_query("SELECT huntId FROM AttendeeHunts WHERE attendId='$id'");

if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
else{
      $rows = array();
      while($r = mysql_fetch_assoc($result)){
            $rows[] = $r;
      }
      header('Content-Type: application/json');
      print json_encode($rows);
}
mysql_close($conn);
```

getMyInfo.php

this file returns a JSON array containing the firstName and id of all the user with the phone number given as an argument

```
<?php

$dbhostname = 'mysql1.cs.clemson.edu';

$dbusername = 'bydtc_bv08';

$dbpassword = 'asdf1234';

$conn = mysql_connect($dbhostname, $dbusername, $dbpassword);

if(!$conn)
{

  die('Could not connect: ' . mysql_error());

}
//echo ('connected');

if (isset ($_GET["phone"]) ){
            $phone = $_GET["phone"];
}
//echo 'MySQL Connected successfully'."<BR>";
mysql_select_db("bydtc_0bhv") or die(mysql_error());

$result = mysql_query("SELECT id, firstName FROM Attendees WHERE
phone='$phone'");

if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
else{
      $rows = array();
      while($r = mysql_fetch_assoc($result)){
            $rows[] = $r;
      }
      header('Content-Type: application/json');
      print json_encode($rows);
}
mysql_close($conn);
```

goingToEvent.php

this file creates an entry in the EventAttendee table with the attendee id and event id given in the arguments, it also updates the enrolled number for the that event in the Events table to the enrolled argument given

```
<?php

$dbhostname = 'mysql1.cs.clemson.edu';

$dbusername = 'bydtc_bv08';

$dbpassword = 'asdf1234';

$conn = mysql_connect($dbhostname, $dbusername, $dbpassword);

if(!$conn)
{

  die('Could not connect: ' . mysql_error());

}
echo ('connected');

if (isset ($_GET["attendeeId"], $_GET["eventId"], $_GET["enrolled"]))
{
            $attendeeId = intval($_GET["attendeeId"]);
            $eventId =intval($_GET["eventId"]);
            $enrolled = intval($_GET["enrolled"]);
}
$errorCount = 0;
if(empty($attendeeId))
{
    echo('Attendee ID empty');
    $errorCount++;
}
if(empty($eventId))
{
      echo('Event ID empty');
      $errorCount++;
}
if($errorCount == 0)
{
echo ('no errors');
//echo 'MySQL Connected successfully'."<BR>";
mysql_select_db("bydtc_0bhv") or die(mysql_error());
```

```
$result = mysql_query("INSERT INTO EventAttendees (eventId, attendId)
 VALUES('$eventId', '$attendeeId')");

if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
else{
      mysql_query("UPDATE Events SET enrolled='$enrolled' WHERE
id='$eventId'");
}
mysql_close($conn);
}
else
{
mysql_close($conn);

}
```

notGoingToEvent.php

this file deletes an entry in the EventAttendee table with the attendee id and event id given in the arguments, it also updates the enrolled number for the that event in the Events table to the enrolled argument given

```
<?php

$dbhostname = 'mysql1.cs.clemson.edu';

$dbusername = 'bydtc_bv08';

$dbpassword = 'asdf1234';

$conn = mysql_connect($dbhostname, $dbusername, $dbpassword);

if(!$conn)
{

  die('Could not connect: ' . mysql_error());

}
echo ('connected');

if (isset ($_GET["attendeeId"], $_GET["eventId"], $_GET["enrolled"]))
{
            $attendeeId = intval($_GET["attendeeId"]);
            $eventId =intval($_GET["eventId"]);
            $enrolled = intval($_GET["enrolled"]);
}
$errorCount = 0;
if(empty($attendeeId))
{
    echo('Attendee ID empty');
    $errorCount++;
}
if(empty($eventId))
{
      echo('Event ID empty');
      $errorCount++;
}
if($errorCount == 0)
{
echo ('no errors');
//echo 'MySQL Connected successfully'."<BR>";
mysql_select_db("bydtc_0bhv") or die(mysql_error());
```

```
$result = mysql_query("DELETE FROM EventAttendees WHERE eventId = $eventId AND
attendId = $attendeeId");

if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
else{
      mysql_query("UPDATE Events SET enrolled='$enrolled' WHERE
id='$eventId'");
}
mysql_close($conn);
}
else
{
mysql_close($conn);

}
```