

Team CSM1

Members: Jake Miller, Lance Perardi, Matthew Wichmann, Sam Elliott

Client: Rob Thompson

Tuesday, June 16, 2014



Introduction

Client Description

The client for this project is Mr. Rob Thompson who is the Assistant Athletic Director/Director of Recreational Sports at the Colorado School of Mines. During the summer he runs a youth sports camp and wants a website in order to replace the paperwork used to keep track of the children at the camp and the guardians responsible for the children.

Product Vision

This product is being designed for use by the Youth Sports Camp. The product stores information on the children that will be attending the camp including guardians that are allowed to pick them up, as well as behavior and accident reports. The user can check a child in and out as well as connect that action to a guardian. Searching for children in the camp by name and group number is possible as well. A report on the children that will be checked out early for the day can also be accessed. This will all be done using a website that is accessible by all devices connected to the Mines network where the data will be held in a MySQL database.

Requirements

Functional Requirements

- search for child by name or group number
- add picture of child and parent
- add notes
- allergies
 - incidents
 - acceptable pick up people
 - early pick-ups and alerts
- ability to check in for the day
- ability to check out for the day
- ability to run reports on database
- import data from an excel spreadsheet
- ability to add in who is allowed to pick up the child

Non-Functional Requirements

- Compressing the pictures so that it does not take up much space on the server
- Check in/out process time needs to be efficient and fast
- Use of an online MySQL database and SQLite android database
- Syncing these two databases
- CCIT will host the server for the website and online database

System Architecture

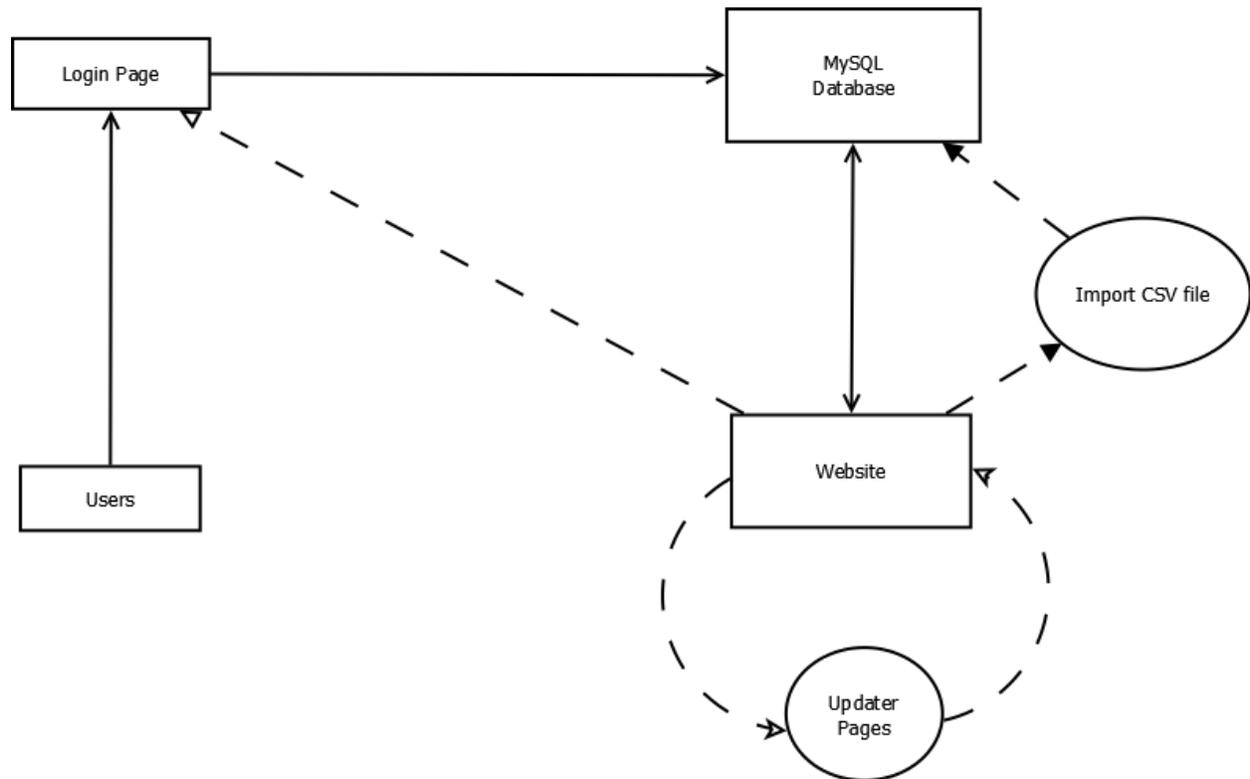


Figure 1: System Architecture

The way users access our website is by being on the CSM network and logging into the website. The users will always be redirected if they are not logged in, meaning that the only page that everyone interacts with is the login page. Once the user has identified them and has sufficient permission to enter the website, which is checked through the "login" table in our database, they then can enter the website. From here, the user can interact in many ways:

- All of the information that is presented on the website comes directly from the database. Each child's information, as well as all the reports and guardian information, is all saved in the database until a user with sufficient permissions clears the tables.
- Whenever a change is made on the website, it is run through a set of update pages, where the page itself changes the values in the database while redirecting the user to the page where the edits were made. This allows us to have the website working in the background to update the specifics that have been changed while the user only sees the end result.
- A user with sufficient permissions may also import a CSV file. This is done through one page, called Import.php, where if the CSV file is in the correct format, then all the information on there will be put into the database correctly. If it is not in the correct format, then the import will not be successful and the database will remain unchanged.

Technical design

Website Map

The way to access the website, as described in the system architecture diagram, is by going through the login page. If you log in properly, then you will come to the Index page, which is the main page for the website. On this page, there are many ways to filter through all the information. Right when you enter this page, all of the children are displayed. You can filter this information by clicking on any of the group/week buttons around the page (shown in the Figure 3).

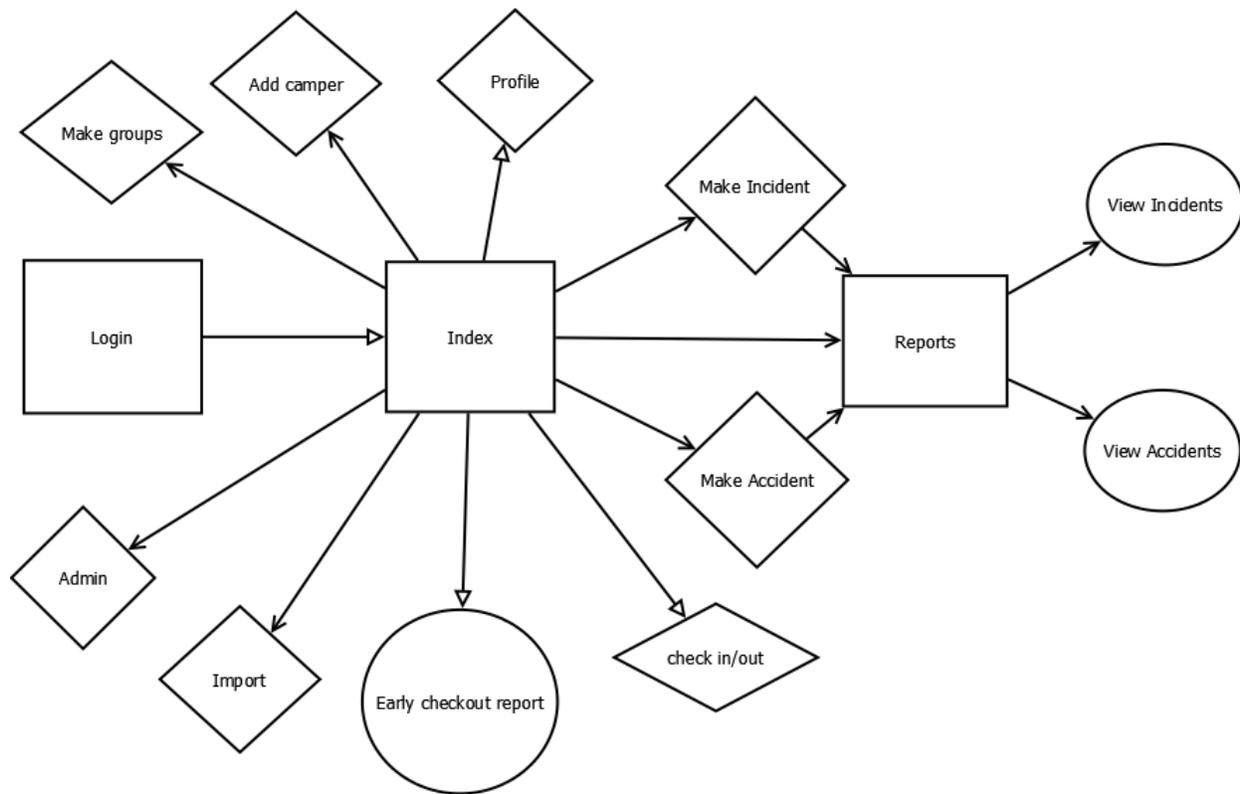


Figure 2: Website Map

On Figure 2, you can see that there are 3 different shapes, squares, diamonds, and circles. Each of these corresponds to the "type" of page it is. The following will explain what each of these is:

- Squares: These are main pages on the website where it is linked back to itself from the links that lead off of it. All of the squares on this diagram show a page that has many links going into it as well as going out of it. With these, there are no editable areas where you can change the database, but you can still access the information, as it is displayed for you.

- **Diamonds:** These are the pages that edit the database directly. With all the different information needed by the client, we needed many different pages to accomplish the tasks and not clutter the user with lots of different information. On each of these pages, there is something that the user can do that will impact the database, being from checking in/out a child to adding in a new incident report to importing new information into the database.
- **Circles:** These pages are similar to the squares, but do not have many different links coming in/out of them. Each of these pages display information from the database, but the user cannot edit them unless they click on the corresponding "edit" button, which will then take them to a "diamond page."

Another major part of this project was that there are different permissions for each user. In Figure 4, there are two different arrows, an open triangle and a normal arrow (similar to this ->). The open triangle shows where everyone can access, being normal and admin users. The normal arrows show where only admins can access, which is all the main sites that can have major effects on the database, such as deleting users from the database or adding in a new camper, for example.

Index Page

Youth Sports Camp

Week

Week 1
Week 2
Week 3

Menu

[Reports](#)

[Import](#)

[Admin](#)

[Add Camper](#)

Group 1

Group 2

Group 3

Group 4

Group 5

Group 6

Group 7

Group 8

[Log Out](#)

#	Child Name	Grade	Group	Get Profile	Guardian Name	Guardian Email	Guardian Phone	T-Shirt
#1	Joe Student	3rd		<input type="button" value="Profile"/>	Steve Student	sstudent@iamcool.com	303-779-5423	Medium (Y)
#2	Sally Ding	4th		<input type="button" value="Profile"/>	Amy Ding	amys.ding@yahoo.com	303-933-1234	Medium (Y)
#3	Jimmy Smithson	1st		<input type="button" value="Profile"/>	Amy Sing	amys.sing@yahoo.com	303-933-1233	Small (Y)

Figure 3: Website Index Page

As you can see in Figure 3, there are many different buttons that you can press, being week 1-3 or groups 1-8. Each of these buttons is designed to be able to filter out the information that is currently displayed on the site so that the user can easily find the camper that they need. There is also search functionality in the top right which you can put in the camper's name and it will display all related campers.

A feature that was implemented was to easily be able to see who has been checked in, checked out, or not yet arrived. The way this was implemented was by color coding the child's

name appropriately. Checked in is labeled as green, checked out is labeled as red, and not at camp yet is labeled as yellow.

Due to the large amount of information and having to scroll to the side to get to the information that you want, there was a highlighting functionality that was implemented. The user can click on a row and it will highlight it as a light blue color, so that the user does not lose which child they are on.

Many-to-many relationship

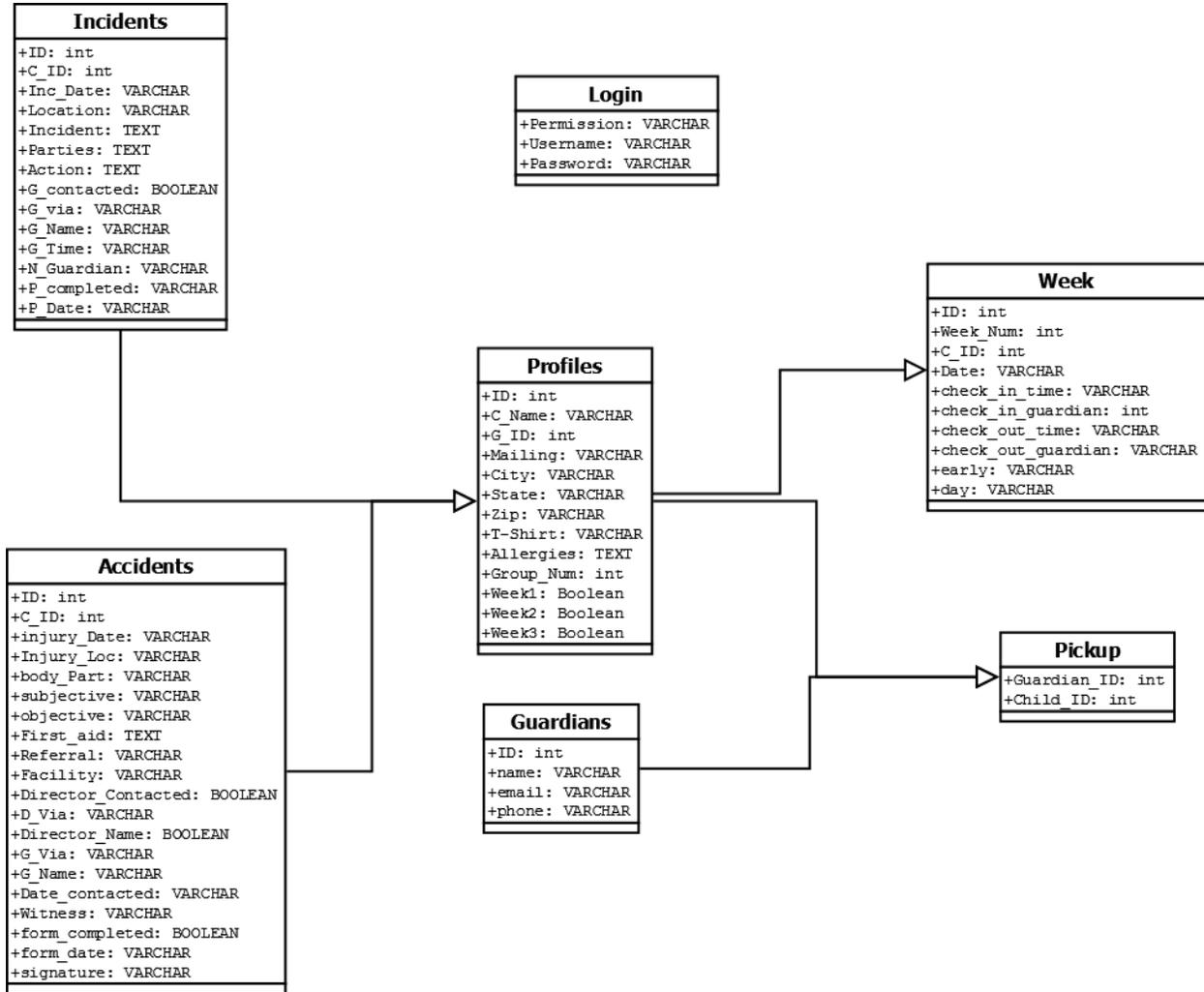


Figure 4: MySQL Database Schema

The client’s description of the product stated that there would be multiple different guardians allowed for pick-up and drop off throughout the course of the camp. For each of the children attending the camp, there will be at least one guardian that will have permission to pick up that child. However, we also had to consider that there are guardians that will be allowed to

pick up multiple different children. So, one child can have many guardians, and one guardian can have many children, creating a many-to-many relationship.

As you can see in Figure 4, the way that we have related the children (being the Profiles table) and the guardians is through the use of a third table called Pickup. This table only holds 2 values, a guardian ID and a child ID. With this, we can easily store all of the guardians in their own table while also being able to call them when we are showing all the child's personal information, which also contains the guardian's information.

MySQL Injection

MySQL injection is a common method used by hackers to gain access to private MySQL databases and potentially make harmful changes. MySQL works by accepting commands called "queries." An example query you might see in code is "SELECT * FROM login WHERE username='\$input'", which would select every entry in the login table where the contents of the username column matches that of the variable \$input, which is set by the user. The returned entry would be used to log the user in. The problem occurs when a hacker sets \$input as "' or '1'=='1' _", which would make the query "SELECT * FROM login WHERE username='' or '1'=='1' '""". Since '1'=='1' is always true, the query will return every entry in the login table, allowing the hacker to log in without a proper username. MySQL injection can do much worse, like using INSERT or DROP queries to manipulate or delete entire databases with little effort.

There are multiple ways to prevent this, like using prepared statements rather than embedding user input in the queries. What we did was escape certain characters in the input like ' and ", prepending backslashes to them. Using the example, \$input would become "'\ or\'1\'=='1\'\'", which would not be accepted as a proper query.

Design Decisions/Lessons Learned

1. Rather than creating both a smaller, less functional website as well as an android application, we focused on creating a complete website that implemented everything (and more) of what the client has asked. Since the website could be accessed from a mobile device, we believed this was a much better solution for the client.
2. We used PHP because the language has a small learning curve so we would be able to learn and implement it in a quick fashion. This language can also be implemented on the CSM network.
3. We created a separate table for the guardians because with this project, the guardians and the children at the camp have a many-to-many relationship. The alternative was cramming the guardians' information into the children's entries, which would inflate the tables and likely repeat a lot of data, so giving the guardian and children separate tables made the database much more efficient and smaller.
4. Another table was created for the check in/out process. Each entry in this table contains the ID of the child being referred to, and who picked them up and who dropped them off on a given day. With this, we are able to display and store every day of the week along with the time that the child was picked up.
5. Due to the many-to-many relationship mentioned above, we implemented another table that will allow the database to link up the children to the guardians allowed to pick up the child. For each child, the table has an entry for each guardian allowed to pick them up, allowing us to search by child.
6. Implementing security for the website and database were a major part of this project.
 - a. Both the website and server were placed inside the Colorado School of Mines firewall, meaning that you have to be logged in and verified on the Mines network to be able to access the internet.
 - b. Due to our database being MySQL, we needed to ensure that the product would be protected from MySQL injection. We implemented certain methods that would stop this from happening and therefore adding more security to the product.

Results

Testing for this project has been done on Firefox, Safari, and Google chrome on both Windows 7 and Mac operating systems as well as mobile devices such as an iPod Touch and an Android tablet. Due to the client's non-use of Internet Explorer, we did not test the product thoroughly with this application. The website can import excel spreadsheets as a CSV file into a MySQL database that relays information to the website so that all the required information that the client wishes to see is displayed. This also has the ability to manipulate the information as the user sees fit. With all the private information provided to the database, we have also implemented security that will prevent those who do not have permission to enter the website from accessing this information.

As a future project, the website could additionally have the capability to store pictures of the children and their guardians, providing more security to the pick-up/drop off process. More functionality could be implemented into the website, such as sorting by each of the columns or a print functionality that will allow the user to print off selected columns for keeping track on paper.

With the database and website now in place, an android application can be developed. The application would have all of the functionality anyone checking in and out children would need in a more interactive and simplistic interface. Other future work that can be done on this product is implementing more advanced security. With the database holding the private information of over 200 children and some their guardians, it would be advisable to implement much more advanced security measures.

Appendix

For both of the spreadsheets, it is very advisable to make it first in excel, then export to a csv file, for more accuracy. For importing users, you can find the link on the admin page. For importing campers, you can go to the import page. Both of the links to these pages are located on the left side.

User Spreadsheet

The following is the format for the spreadsheet that is needed for the importing users to work:

	Name	Username	Password	Permission
--	------	----------	----------	------------

A-1: User Importing Format

The first column, being the blank one, will be put in column A in the spreadsheet. In permission, the value must be a 0 or a 1. 0 corresponds to being a normal user, 1 being an admin. Most likely, for the client, the only person who will have admin access is him, which he has been put into the database already, so that he can just add in everyone he wants.

Once all the information has been put into this spreadsheet, the user with admin access should go to the admin page; the link is located on the right side of the website, and click on the import button. This will allow the user to put other users into the database so that more people, being the employees of the camp, which will give them access to the website.

Camper Spreadsheet

For importing a set of campers into the database, the format should be the exact same as the Excel spreadsheet that the client, being Rob Thompson, gets from CashNet. The following is the spreadsheet format to be followed. Below, there is the “Column on Spreadsheet” which is the letter of the column that is being used for the “Information” that is in the table.

Column on Spreadsheet	Information
O	First and last name
T	Parent/Guardian Name
Z	Parent/Guardian E-mail
AD	Parent/Guardian Phone
AH	Primary First Name
AJ	Primary Last Name
AL	Primary E-Mail
AP	Secondary First Name
AR	Secondary Last Name
AT	Secondary E-Mail
AV	Secondary Phone
AX	First Name
AZ	Last Name
BB	Email
BD	Phone
BF	First Name
BH	Last Name
BJ	Email
BL	Phone
BN	Mailing Address
BP	City
BR	State
BT	Zip Code
BV	T-Shirt Size
BX	Grade for upcoming year
BZ	Please group with (one name only)
CB	Allergies (if none, please state N/A)
CD	June 2-6, 2014: \$260.00
CF	June 16-20, 2014: \$260.00
CH	July 14-18, 2014: \$260.00
CJ	Amount

A-2: Importing Children Excel format

User manual:

Checking campers in/out:

1. Select the week that you are currently in for the week. These buttons are located on the top of the page. You can also click on the group buttons located on the left side to narrow down your search.
2. Select the child that you want to check in/out. This is located next to the child's name on the table.
3. Click the button named "Check in/out"
4. You will come to a page that looks similar to the following figure:

CHECK IN AND OUT FOR WEEK 1 ON Wednesday

Steve Student 303-779-5423 ▼ Checked In OTHER:

Early Check Out Information

Time:

Short explanation:

Steve Student 303-779-5423 ▼ Checked Out OTHER:

A-3: Check in/out Page

5. Here you can select which option to either check in or out, depending on what the user wants to do:
 - a. With both, you can pick the parent/guardian that is checking in/out the child.
 - b. If there is a new parent/guardian then you can put it under the OTHER section.
 - c. For early checkout, the user can put in the time that they will be leaving along with a short explanation.

Making reports for a child:

1. Go to the home page. Click on the home button if you are on a group page.
2. Select the week that the child is in. Afterwards, find the child that you want.
3. In the child's row, click on the type of report that you want to create.
4. Fill in the areas that need to be completed and hit submit.

Permissions for the admin users:

For being an admin user, you can do many other things besides making reports and checking in/out campers. The following walks you through how to access different areas on the website that will allow you to do certain actions:

- **Admin Page:** Here, the admin users can import new users for the website. Please refer to A-1 for instructions on how to import these users. On this page, you can also delete current users and see their information.
- **Add Camper:** This page is used for adding in campers that were not on the original spreadsheet. It has all the same information as on the profile page for a camper so that all this information can be put in.
- **Reports:** Here you can view all the reports that have been put in over the span of the camps.
- **Early Checkouts:** This is where an admin can view all the checkouts that are happening during the week so that s/he can be prepared for when all the campers that are leaving early from camp.