Events Management

Isabel Alvarez
University of Virginia
ia2ew@virginia.edu

Jake Morrison
University of Virginia
jlm8ps@virginia.edu

Caymen Rexrode
University of Virginia
cnr7ka@virginia.edu

Louis Thomas
University of Virginia
lat9nq@virginia.edu

Matthew Vick
University of Virginia
mev8vy@virginia.edu

Jeremy Weng
University of Virginia
jhw2np@virginia.edu

ABSTRACT

The Virginia 4-H State Shooting Education Program provides hands-on opportunities for youth to learn responsibility and discipline relating to firearms. Competitions held by the program require hours of planning and labor. Our goal was to create a system which improves the overall efficiency and simplicity of this process. The system provides an interface accessible to the head of the program, administrators of the competition, and coaches which can be used to register participants. Additionally, the system automates a previously manual and tedious process of scheduling each individual player for requested events within a competition.

ACM Reference Format:
Isabel Alvarez, Jake Morrison, Caymen Rexrode, Louis Thomas, Matthew Vick, and Jeremy Weng. 2019. Events Management. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The goal of the Virginia 4-H State Shooting Education Program is to use firearms handling and competitions as a means to teach youth from ages 9 to 19 across the state of Virginia important life skills. Through one of several shooting disciplines ranging from archery to rifling, the program seeks to instill participants with responsibility, teamwork, and discipline, as well as teach firearm safety. The Program runs multiple competitions throughout the year, including an annual State Shoot.

Youths are enrolled in these competitions by their parents or coaches, who must personally contact the head of the program, and provide a list of players and specific relays (i.e. pistol, rifle, bow, etc.) their players would like to participate in, and what days they would be available to participate (the competitions run on Saturday and Sunday). The organizers manually enter this information into a spreadsheet and sort through players to create a schedule of events that attempts to group as many players together as possible for the requested events and the days the players are able to participate. Complicating matters further is that some events divide players based on their age group. As one can imagine, this process is lengthy and tedious, especially when many coaches and players are involved, and also when coaches have specific requests for how their players are assigned to events. For example, two players may share weapons and so must be arranged in the schedule so that they do not participate in simultaneous events, or two siblings may request to participate in the same event at the same time. The head of the program has much experience in how to navigate this complex process, but feels that this experience must be streamlined in order to help him accomplish his duty, and to help successors to his duty be just as effective without having the experience he has. Therefore, our project consists of creating a web based application specifically designed to incorporate and streamline all aspects of this process, from enrollment of coaches and players to scheduling.

Our system will benefit the organizers by significantly decreasing the amount of time they spend on enrolling participants. This process will also eliminate the potential for user error by constraining the type of information that can be entered. Since the scheduling has been mostly automated, we have removed the vast majority of time that the organizer must spend manually preparing a certain competition while also further reducing the possibility of human errors. The process will also allow coaches and parents to become more involved in the process and reduce their potential for user errors. Due to the reduced dependence on emails between competition organizers and coaches in our system, the overall amount of time spent enrolling participants will be reduced for all parties.

One of the most important benefits of our system is that it will allow any successor to the current program head to perform these duties effectively without needing extensive knowledge of the current scheduling procedures. Over the years, the program's head and his team have developed a very specific set of procedures and knowledge that may not transfer easily to a successor. Since the format of entry has changed from one dependent on specific procedures and format to a more general one, any user can easily pick up the system and accomplish the duty of the program's administrators. This will leave the program in capable hands in the event that its current head retires.

2 BACKGROUND

In order to accomplish the creation of the Events Management system, the team leveraged a number of tools, frameworks and languages that provided the best results. For all parts of our project with the exception of the automated scheduler, we wrote the code in Python and HTML using the Django framework. Django implements a Model-View-Controller framework, with a SQLite database as the model, HTML templates as the view, and Python code as the controller. Our templates additionally required the use of some

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Javascript in order to render information and form elements. The appearance of the user interface was improved and made more accessible to users by using the Bootstrap CSS library. In order to implement the automated scheduler, we designed a C program, which we embedded into our site for increased efficiency.

At the request of the head of the shooting education program, the team is working on deploying the system without using a cloud provider. It was preferred that the system was deployed for production by using a server administered by a person of the program head's choosing due to costs and security reasons. Said server administrator will be tasked with handling any system maintenance needs after our team has delivered the completed product.

3 RELATED WORK

There are a number of systems dedicated to the scheduling of events, appointments, classes, visits and such. A rather comprehensive list has been compiled by a third party company and can be found here https://www.capterra.com/scheduling-software [1]. Most of these systems are designed to fulfill the specific needs of the industry that they aim to reach. For example, doctors offices use scheduling software that allows a user to pick from a number of a available slots [2], while restaurants use this type of software to ensure that there will be an appropriate number of workers in the establishment at all times [3].

The systems that are readily available to consumers are usually generic, and have limited customization options. They are designed to fit the needs of a wide range of users, and therefore cannot accommodate for specific requests. While custom solutions are certainly also available, the existing ones have been tailored to another user's particular needs. If the head of the shooting program had chosen to procure a system that was specifically adapted to his needs, this would have likely been costly and time-consuming.

While these off-the-shelf solutions are certainly effective in many cases, they have a number of drawbacks. Firstly, almost every one would have involved some cost to the shooting program. Additionally, designing a custom system from scratch as our team has done would have been much more expensive if the head of the program had chosen to pay for the services of another team of software developers. The pre-made systems would have not been acceptable, due to the fact that the head of the program also wanted the system to use specific language and meet specific requirements, which an off-the-shelf system would not have been able to achieve.

Our team chose to develop a custom built system that would better accommodate the specific needs of the 4-H Shooting Education Program. By working closely with the head of the shooting program, we were able to create a system that is specifically tailored to solving the problem at hand. The system uses the appropriate terminology, it can limit access and visibility of player information based on the role of the logged in user, and it provides the authority to run the automated scheduling algorithm only to those users with an administrator role. These and other specific features that our system implements make it a better choice than any of the other systems available in the market.

4 SYSTEM DESIGN

4.1 Framework

When presented with the problem, our team decided that building a web application would result in the best results, since this would reduce the barriers to entry that the users would encounter by removing the need for them to install extra software on their machines. The application was built using the Django framework. The team found that this framework was powerful enough to provide us with the right level of support, and at the same time relatively easy to learn and troubleshoot. The team also took into account in this decision the fact that there is abundant documentation online, as well as a wide network of developers working with this framework.

4.2 System Elements

When designing the system, the team was mindful to use vocabulary that was aligned with the terms used in the shooting education program. By following the program head's guidelines, we designed a number of objects in the system. For instance, the annual Virginia State Shoot would be represented by a Competition object in the system. Associated with Competitions are Events, such as archery, PPP air pistol, skeet and trap. Athletes - the children participating in the competition - can choose which events they are interested in. Each Athlete is associated with the Club they belong to. There is usually one Club per physical county in the state of Virginia, although there are exceptions. For instance, some "Clubs" are merely parents enrolling their children in competitions rather than a larger and more extensive group. Each Club designates one or more adults to assume the role of Coaches. The Administrator of the system will grant access to those Coaches, who in turn will have the responsibility of entering the Athletes in their Club into the system, and listing which of the events they are interested in. Once the majority of the Athletes have been recorded, the Administrator will be able to run the automatic scheduler, which will generate enough Relays to accommodate all the Athletes registered for that Competition at the time. A Relay is an occurrence of a certain Event; for instance, a 45 to 60 minute slot where a group of Athletes will be shooting at the same range at the Competition site. By using this terminology, the team aims to make the transition into the new system easier for the administrators and the other users.

4.3 Users

Two key types of user exist in our system: Administrators (Admins for short) and Coaches. Each of these user types has access to their own portion of the site (they are redirected to the appropriate section when logging in), and both are capable of interacting with the database. Admins are the role assumed by the team in charge of the Virginia 4H Shooting Education Program right now. Their most important function within our system is that they can create and modify objects corresponding to Competitions, Clubs, Relays, and Events. They are also capable of creating County objects and accounts for Coaches. In addition, they are able to create and modify Athletes for all Counties and Coaches, as well as alter the relays they are enrolled in. The responsibility of starting the automatic scheduler, adjusting the results, and generating most reports falls to the Administrators. They are also capable of creating other Admin users with full access to the system.

Coach accounts fill the role currently assumed by coaches for the various clubs that enter the State Shoot. Coaches are solely responsible for creating and editing information for Athletes within their county. They are not able to edit any information about their county or directly schedule their Athletes for events. Instead, they simply enter the type of event each Athlete wishes to enroll in and the automatic scheduler takes care of the rest once the majority of players are enrolled. Any special requests for relay accommodations are sent directly to the Admins via email. We decided on this due to special requests being infrequent enough to not warrant its inclusion, in addition to the original system being a better way to ensure administrators were notified of the request. Coaches are also not able to create other Coach accounts in the system- instead, requests for accounts are submitted via email.

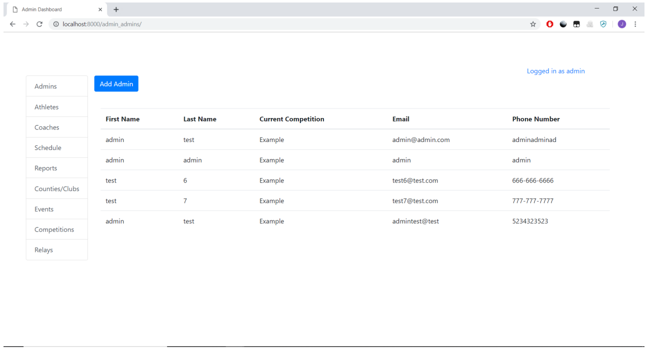
4.4 Key Functionality

The process of enrolling a participant in a Competition consists of a representative of a County or Club (referred to as a Coach for simplicity) contacting an Administrator (the role currently filled by the program head and his associates) with a list of Athletes and the Events they wish to enroll in. The Competitions themselves consist of multiple Relays - iterations of a specific type of Event.

The system was designed to relieve the admin of the daunting task of entering all the athletes’ information. This is accomplished by enabling the coaches of each club to enter their own players’ details. The coaches are also able to generate a report containing the schedule of the players in their club. By separating these tasks and allowing coach users to perform them, the system relieves the administrator’s workload. This also allows coaches to feel more invested in the process and ensures that they know definitively that information is being received by the administrators.

Particularly in the admin interface, the user interface is designed to be as clear and intuitive as possible. The tabs along the left side of the screen replicate the main types of objects that exist in the system. Within each tab, a table showing the relevant information for the type of object, be it athletes, coaches, events or relays. Each row in said table will represent an instance of that type of object. To the right side of each row there are Edit and Remove links that allow the user to take actions that are specific for a single instance of a certain object type. Also at the top of each page there are buttons that allow the user to take the actions that are relevant to the object type displayed in that page. All these elements can be seen in the screen-shot of the administrative interface found in Figure 1.

Another key feature of our project is the creation of reports. Both types of users are able to generate and print lists of players and events in the current competition. Administrators are also able to view reports of all counties/clubs involved in the competition. These reports, in particular the ones the administrators have access to, were implemented in a format that would make it easier for the users to visualize the information. They allow for the information about players’ schedules to be extracted from the system in such a way that the administrator can simply send the downloaded PDF files to the interested coaches and other event staff.



First Name	Last Name	Current Competition	Email	Phone Number
admin	test	Example	admin@admin.com	adminadminad
admin	admin	Example	admin	admin
test	6	Example	test6@test.com	666-666-6666
test	7	Example	test7@test.com	777-777-7777
admin	test	Example	admintest@test	5234323523

Figure 1: Administrative dashboard of the Events Management web application

Additionally, per the program head’s request, the team built a CSV reporting functionality alongside the regular PDF reports. This CSV report will serve the purpose of allowing the administrator to export the majority of the data from the system and into a tool of his choice. The program head has been using Microsoft Excel to schedule these shooting events for several years, and although the software the team has developed has replaced the scheduling function of these Excel sheets, the organizers will still need to use external tools to keep track of catering, lodging, and other costs and fees associated with the competitions. Rather than implementing these functions in our own system, which would have been our of the scope of the problem we set out to solve, the team decided to enable the organizers to export the information gathered by the system and put it to use in a tool of their choice to accomplish these other secondary tasks.

After the automatic scheduler is executed, a number of relays will have been created and athletes will have been assigned to each. Although these relays are visible in a list view from the main admin interface, the team wanted to provide the administrator with a more intuitive way of visualizing the way these relays are scheduled by using a calendar view. We integrated the Google Calendar API into our system such that every relay gets automatically converted into an individual event in this Google Calendar. This calendar is visible from a page in the administrator interface. Optionally, due to the fact that the calendar created is a standard Google Calendar, the administrator could share it with those who need access to all the information contained in it. Overall, this feature allows for yet another way to visualize and share the schedule that has been automatically created by the system.

4.5 Scheduler

Scheduling is a critical part of the program head and his team’s process when setting up competitions, and we had much to consider when planning how to implement it. As mentioned previously, coaches or parents will often make special requests to the administrators asking for special arrangements or alterations to the scheduling of participants. The competition’s schedule is required to leave plenty of leeway in order to accommodate these changes, as well as provide space between relays in order to allow participants and staff adequate time to move between venues, leave adequate time for participants and staff to eat lunch, and ensure that there are enough relays to allow all participants to compete before Sunday afternoon so that the staff can tally scores and announce winners in the early afternoon. To accomplish this the scheduler initially schedules all of the events one after another. It keeps track of which slots belong to which day, and then, in a post-processing routine, the scheduler maps each event to an actual time during the day, leaving a break for both lunches and between days.

The scheduler is designed to use a greedy algorithm to determine athlete relay placements. This part of the project was mostly designed separately from the Django components of the application. The main link between the scheduler and the rest of the application is an SQLite 3 database. The scheduler opens the database and reads competition, event, player, and player-event association data, as well as clears previous schedules. Competition data helps the scheduler keep specific to relevant competition data and avoid older or future competitions. Player-event associations are a result of the many-to-many relationship in the database. The scheduler sorts player-event associations first by player availability (only day two then only day one), then by players with more events signed up. The list is then mapped to a series of slotted times. Multiple events may take place in a single slot, but each event in a given slot must be unique. The mapped slot times make up the schedule, which is then formatted and forwarded back to the original database.

4.6 Challenging Design Decisions

The scheduler uses the C programming languages for its performance improvements over other available languages. When it was time to start designing the component, it was noted that the scheduler did not need to be integrated into the whole system. Rather than doing so through the abstractions provided by Django, the scheduler could interface with the database directly. It was also unknown what the run-time expectations of the scheduler would be, so a programming language with good performance was needed to ensure that a schedule could be successfully made. The scheduler needed a set of extra kernel-level functions in order to prevent blocking. Since the database's output never makes it clear when the output of a query ends, the parent would block if it tried to read it directly. Utilizing forks and `pthreads` allows the scheduler to open the database once and query it multiple times during a single session.

Another challenging design decision was whether or not to export reports in the Excel format the administrators are currently familiar with. The administrators expressed their desire to export the data that had already been collected by the system in order to use it in other miscellaneous competition-related tasks. However, this would be more difficult using the PDF format that was used for all the other reports. Ultimately, we decided to export the files as CSV file instead, due to the fact that these types of files are more universally accepted by editors and viewers that the administrators might be using to process this information.

5 RESULTS

Ultimately, our software resulted in a massive increase in productivity for the organizers. While creating a schedule would normally take an experienced organizer several hours (usually around 2 or 3), our scheduler speeds up the process to just 2 minutes. Overall, many processes that required upwards of 10 to 20 minutes, such as the addition of players to the roster, now take around 2 to 5. Because of this and the overall ease of use of our software, the head of the program and his team have used it to fully replace the former system of Excel databases.

While coaches/parents of prospective competitors have not had a chance to use the software yet due to our product not being deployed for use in enrollment for a competition, the program head believes that they will be satisfied with the greater degree of control they have in the process. The stakeholders are very satisfied with the software, and feel that it is accessible enough to allow future event organizers to accomplish their duties with little to no issues.

In synthetic tests of the scheduler, 400 athletes participating in seven different events were generated and added to the database, with a random number of many-to-many relations linking the two, typically around 1,800. A schedule of relays was then created, and `time` was ran to benchmark the scheduler. In a test on Ubuntu 18.10 with an Intel Xeon E5-2630 v2, the scheduler takes less than 9 seconds to download the data and schedule it. Real time was reported to be 8.862 seconds, while user-space CPU time was 0.076 seconds and kernel-space CPU time was 0.268 seconds, suggesting that most of the scheduler's time was spent reading data from the database (though this claim is not thoroughly verified). The large amount of real time is due to the non-blocking method used to read the data: it waits a full second each time it sends a query and expects output from the database.

6 CONCLUSIONS

We designed a system to manage and schedule events for the Virginia 4-H State Shooting Education Program. The system was built using the Django Model-View-Controller framework and code written in Python and HTML with the exception of the scheduler which is a C program. Using C for the scheduler increased the overall efficiency of the system. Additionally, it is essential for the system to match the language used by the shooting education program. Language consistency provides a decreased learning curve and limits the potential for errors within the system caused by misinterpretation. Our system increases productivity for program organizers by automating scheduling, the most time consuming portion of the event creation process. In addition to increasing time efficiency, our system gives coaches and parents more direct and increased access throughout the sign-up process for an event.

7 FUTURE WORK

Our system was designed mainly to address the needs of the 4-H Shooting Education Program as they pertain to the annual State Shoot competition. However, not all of the competitions the Virginia 4H Shooting Education Program runs are of the same format. Notably, some competitions may be longer or shorter than the two days the State Shoot lasts. In the future, our software could be adjusted to allow for competitions of a more variable format in order to address all possible competitions. This task was out of the scope of this year's work due to the State Shoot being our primary concern and the difficulties associated with adjusting the scheduler and other elements of our software to work with more or less than two days of competitions.

Another subject for future work would be scoring for the events. As this is also a lengthy and tedious process for the organizer and State Shoot staff, the program head expressed a desire for us to integrate an external scoring software into our own. Although this task was out of the scope of our team's work, it would add great value to the system, moving it in the direction of being a centralized place of record for all information regrading the shooting competitions. This would in turn tie into another area for future work: converting the current system an accessible database that would allow coaches and administrators to track the performance of competitors across multiple competitions.

8 ACKNOWLEDGMENTS

Our team would like to thank our customer, Paul Benneche, for all his invaluable help throughout the process of designing and developing this system. Without his feedback, the system would not have been as well suited to the needs of the Virginia 4-H Shooting Education Program.

REFERENCES

[1] Best Scheduling Software | 2019 Reviews of the Most Popular Systems. (n.d.). Retrieved March 31, 2019, from https://www.capterra.com/scheduling-software/

[2] Online Appointment Manager Reviews and Pricing - 2019. (n.d.). Retrieved March 31, 2019, from https://www.capterra.com/p/14916/Online-Appointment-Manager/

[3] 7shifts. (n.d.). Employee Scheduling Software For Restaurants. Retrieved March 31, 2019, from 7shifts website: https://www.7shifts.com/employee-scheduling-software-restaurants