



**University of Arkansas – CSCE Department  
Capstone I – Preliminary Proposal – Fall 2021**

**Fayetteville Public Library  
Volunteer Time-Tracking and Management System**

**Kagen Crouch, Steven Trinh, Alexis Carter, Bradley Lithalansy, Nathan Secrest**

**Abstract**

The Fayetteville Public Library provides opportunities for individuals looking to make an impact within and around our community. Our group has been tasked to revamp the system currently in place that is responsible for the organization and management of these volunteers. To accomplish this task, our group has been asked to create a new application decentralized from the library's appropriate departments that no longer requires individuals to no longer check in through the library's systems in house.

This application is to be mobile, and web based. It will also contain the following features: volunteer check in and check out, new user registration and relative volunteer assignments. This application will also work in accordance with the library's current database. This gives the library access to volunteers' personal information such as but not limited to their name, date of birth and their current volunteer assignments. This application's goal is to make the volunteer process of the public library very low friction. It is also our hope to expand this application into the Northwest Arkansas region so that other communities can have effective volunteer management systems.

**1.0 Problem**

The Fayetteville Public Library has an average of 220 volunteers working on a variety of projects both in and out of the community, to help support the library's mission of strengthening the community and empowering citizens with free and public access to knowledge. The problem the library wants us to solve is to create a new volunteer time-tracking and management system to help quantify the impact their volunteers make on library programs and events, along with the positive impact made on patrons' experiences at the library.

The impact made by volunteers on the library's quality and variety of programming offered is highly valued and, as such, they hope that they will be able to quantify such impacts and improve the quality of life for the volunteers with the new management and tracking system. Moreover, they would like to see capability for the app to be white labeled so that it may be used by other non-profits and libraries that have similar needs for volunteer tracking, which would benefit more communities. Without a solution to the library's problem, they will lack a modern, proper way of quantifying the impact of volunteers on their programs

## **2.0 Objective**

The objective of this project is to build a modern application that is web, Android and iOS based that will improve the volunteer experience for both at the library and off-site. The application should incorporate Human Centric Design for ease of use, geofencing, and the ability to use QR codes for volunteers to check-in or out of a project, task, or department. It should also coordinate large scale events and a large volunteer base.

The management side of the application should allow for a variety of reporting options, such as uploading “canned” reports and having the ability to create custom reports on different projects or overall time reporting. It should be friendly to non-technical users and allow for individual volunteer, project additions, and deletions. For more technical users, it should allow bulk additions and removals of users and user information. The app should have the ability to group volunteers and export their email addresses for bulk emailing. The implementation of the database, admin user interface, web and mobile API should be open-source tech and cloud-based.

As a stretch goal, the app should have the ability to white label so that it may be offered to other non-profits and libraries in the region, should they have similar needs for the tracking of volunteers.

## **3.0 Background**

### **3.1 Key Concepts**

#### **Python Django**

Django is a high-level python web-framework used to create web apps rapidly and smoothly. Django is a free and open-source framework built by experienced web developers, famous for its ability to scale to larger needs, making it ideal for our use case. Another tentpole feature of Django is its security, protecting against various forms of attack such as SQL injections and cross-site scripting to name a few. This feature is valuable to the implementation of this project due to the systems need to house personal information about volunteers using the system. If our system was not to have protection against various forms of attack, personal information may be at risk.

#### **Flutter**

Google’s User Interface toolkit that uses Dart as its programming language. Flutter is open-source and used as a convenient tool to create cross-platform applications for both mobile, web, and embedded devices. With it being developed by Google, it comes equipped with functionality that is used with translators such as Google Translate, voice recognition software such as Google Assistant, and merchant apps that read credit card information to make purchases. It allows you to create a native mobile application with only one codebase, meaning that it can be used to create the same app for both iOS and Android, for example. It compiles ARM, Intel machine code, and JavaScript to optimize time spent compiling. For similar reasons, it also has automated testing and developer tooling.

#### **Material Design Google**

Material Design is a design language created by google in 2014. This system is used to flesh out the designs of a mobile app. Material Design uses grid-based layouts, animations, and transitions to create a consistent and fluent user interface that is also especially pleasing. Material Design uses designs that reflect the physical world, specifically the medium of paper and ink. It also maintains a hierarchy within its elements to create a focus for the viewer's experience. The system will help serve the functionality of the website while not taking valuable features at the same time. In addition, Material Design has a built-in state system that has component libraries available for Android, iOS, and Flutter. Those components have the potential to cover display, navigation, actions, input, and commination. If necessary, Material Design also has a theming option that allows for a reliable consistency throughout the system.

### **PostgreSQL**

PostgreSQL is an open-source object-oriented database that has been in circulation for over 30 years. Its tentpole features being reliability, and performance while extending the SQL language. PostgreSQL helps administrations protect data integrity and build fault-tolerant environments within their databases. It allows for original data types, custom functions, and can be used in different programming languages without recompiling. It also has defined APIs for many features for ease of use and is highly scalable. This software will be invaluable to the public library on protecting essential volunteer information secure and private. Ubuntu is an operating system that will interact with our PostgreSQL database. Ubuntu is also a free and open-source software.

### **3.2 Related Work**

GivePulse is a volunteer tracking website used by the University of Arkansas and various clubs within it to track volunteer hours as well as participation in events. It allows volunteers to clock in/clock out of events. People can register to volunteer at an event ahead of time, which is useful to see who will be participating and if there will be enough people. GivePulse specifically allows students at the University of Arkansas to log in with an SSO (Single Sign On), which allows people to sign in only once on a device and have access to everything without having to type credentials again. This makes it easier for volunteers to log in and register for events; if a person is at a remote event without computer access, they can log in on their phone and clock in/out their volunteer hours. This service also allows event organizers to keep track of no shows and make notes of certain volunteers. GivePulse allows people to communicate via message boards, advertise outreach opportunities, and approve and edit submitted projects, events, and opportunities.

A problem with GivePulse is that it is not tailored for library use, so it cannot access the library databases. More specifically, we would be unable to import all the existing volunteer information into GivePulse, so volunteers would have to create their own account on the GivePulse website and old information would be lost. With our solution, we have the option to import user information from the old database to our new database.

We want our app to allow users to connect with the library's database. Our app would also be open source and contain documentation so the framework could be used by other libraries and expanded upon. GivePulse is proprietary software, and it would cost the library \$138.00 a month to use. This makes it imperfect for our project requirements. Our system will be free to use to encourage other libraries or organizations in the north-west Arkansas area to use the system in

their volunteer services. Being open sourced would allow other libraries to connect their own databases to our system as seamlessly as possible.

## **4.0 Design**

### **4.1 Requirements and/or Use Cases and/or Design Goals**

- Web and mobile apps will have a login screen where new users can create a login and existing users can login with their own credentials.
  - New users will need to be approved before gaining access to the functionalities of the app.
- Web and mobile apps will allow users to enroll in volunteer opportunities given by the library.
  - Once a volunteer has enrolled, they will be expected to check in to the event, through either a QR code volunteers will scan and input their information to confirm their presence, or a sign in screen built within the system.
  - After the event has concluded or the volunteer has signed out of the event the system would then compare their log on and log out times and then store this information within the database to log their hours.
- Mobile app will have the option to scan a QR code that will automatically log the time for the volunteer.
- Web/mobile application will allow admin users to manage volunteers and their time without having to enter the database directly.
  - Add and or remove volunteer hours to specified individuals
  - Add or remove volunteers
- Mobile app will have a simple interface to not confuse the user on its functionality.
- Web applications will have a reporting system that will allow admin users to generate various reports on volunteers and their hours.
- Volunteers will be able to log their hours outside of the library using the mobile app. In combination with a QR code.
  - Offsite volunteer opportunities will require a QR scan to check in to events. This is to prevent incorrect logins of volunteer's times when the volunteers are not present
- Volunteer information and hours will be logged into the existing database in the library.
- Volunteers will also have access to their personal information and volunteer hours currently logged through a personal volunteer page within the web and mobile app.
- A Whitelisted version of the above systems described is also to be created to allow other organizations to use these systems as well
  - This version would then be tailored to the organization to fine tune it for their current and future needs

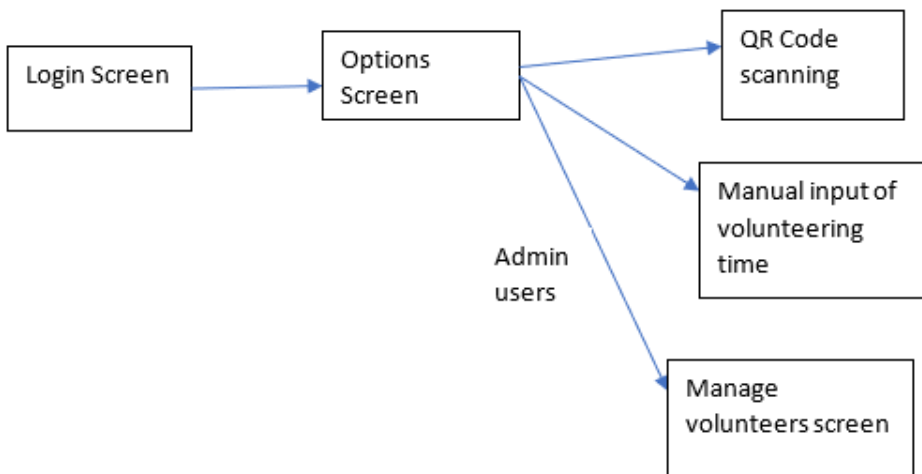
## 4.2 High Level Architecture

The major software components of this project are Python Django, Flutter, Ubuntu PSQL, and Material Design by Google. Flutter, Material Design and Python Django will be used to implement the front-end. Ubuntu PSQL will be used to implement the database. The application will be made for both Android and iOS devices.

Application Goals & Features Overview:

- Being able to track the contributions of library volunteers
- User-friendliness (Human Centric Design)
- QR codes for volunteers to check-in or out of a project, task, or department
- The ability to create both canned and custom reports on volunteers' time contributions to projects
- Coordinating events for the library
- Bulk emailing
- Managing volunteers
- Mobile API to be built with open-source, cloud-based technology

Application Front-end Design Diagram:



The application will have limited functionality for the volunteer-level users. Volunteers will be able to login and clock in/out either manually or with a QR code. The library, or other organizations, admins using the system will be able to post volunteer events. Once events are posted users will be able to select from the posted opportunity and then register for opportunities that align with their availability. This makes the system more independent for the library staff, and admins monitoring the system spend less time having to find volunteers with availability.

Admin users will be able to manage volunteer information directly. These features will allow admins to add or remove volunteer hours as they see fit and allow for the addition or removal of volunteers from their system. Admins will also have options that allow them to notify volunteers of current and future events. This is important in making the system more independent for the library staff having to manage individuals and their ranging availability. The

app will be created using Flutter to ensure that it is cross-platform between both iOS and Android. We want to make sure that the app is simple and intuitive as possible, so the app will need to be designed with appealing icons that do not need much explanation for their use. This is where Material Design is useful as it comes with many different components that we can integrate into our Flutter program. Some of these components include buttons and different layouts that we can set up for the different screens being displayed. However, the system will be built with future updates in mind so that others using the system can add or remove functionality as they see fit, such as later adding another option for admin users to build and print reports from their phone.

The web application will be built with the existing website that is currently in use in mind for tracking volunteer times. It will be implemented using Python and Django and maintain the same functionality by allowing volunteers to login and manage their volunteering times. The improvements will be on the administrative side that will allow the admin user to generate various reports based on the volunteer information. The idea is to redirect the admin user to a page with various options that they can select from. These options will include a dropdown menu of different reports they can view/print of volunteers and their information, including their volunteering time. From the options page, the admin user can also be brought to another page that will allow them to manage volunteers and their times. The page will display a list of volunteers in a table format and there will either be options above the table for editing or we will make it so that the table itself is editable. The list will contain volunteer information such as their name, date of birth, picture, volunteer times, etc. We will ensure that this information is kept confidential by keeping passwords hashed in the database and allowing only admin users to access this page. There is currently a PostgreSQL database that the library uses to keep the information of the volunteers, however we can create a new schema within the database to keep additional information if the need arises.

### 4.3 Risks

Risk	Risk Reduction
Breach of sensitive information	<ul style="list-style-type: none"> <li>- Hash user passwords and use secure programming practices</li> </ul>
Bad database design for storing user info	<ul style="list-style-type: none"> <li>- Provide documentation for database</li> <li>- Normalize data</li> <li>- Choose proper primary keys</li> <li>- Use good naming conventions for attributes</li> </ul>
Hard to maintain code/disorganization between team	<ul style="list-style-type: none"> <li>- Provide documentation so team members and future developers can understand the code</li> <li>- Use GitHub for collaboration and version control</li> <li>- Variables and functions should be named well, and there should be comments to explain pieces of code</li> <li>- Use interfaces and abstract classes to secure data</li> </ul>

#### 4.4 Tasks

1. Visit the library to understand the process needed to be currently taken by volunteers.
2. Understand how the library currently organizes its systems
3. Discuss and design the high-level view and expected features of the application with sponsors
4. Learn Flutter and how to build applications with it.
5. Begin the implementation of front end, bare bone implementation
  - 5.1 Create login page
  - 5.2 Create volunteer and admin areas
  - 5.3 Begin to build user functionality such as inputting personal information and availability
  - 5.4 Create admin functions such as adding and removing volunteers and hours for individuals.
  - 5.5 Create the ability for admins to add and remove events
  - 5.6 Begin to implement volunteer registration for events
6. Begin the implementation of back end
  - 6.1 Create functions that pull information from database
  - 6.2 Build a connector between the front-end code and back end
  - 6.3 Build functions that are ready to receive information from the front end
7. Align existing library volunteer database with back-end implementation
8. Finish implementations of front end of website
  - 8.1 Finalize admin area and functionality
  - 8.2 Finish implementation of the user area and functionality
9. Test back end of the application
  - 9.1 Ensure insertions, deletions, and updates to database can occur
  - 9.2 Check that Insertion, Update, and Deletion Anomalies do not occur to confirm that our database normalization is correct
10. Link together front end and back end containing database
  - 10.1 Connect events in the app (such as login button being pressed, or create account button) so that they query the database based on user inputs
11. Test system as a whole
  - 11.1 Test cases such as:
    - 11.1.1 User tries to an account with a username that already exists. Does the app prevent it from happening or does it overwrite the first account created?
    - 11.1.2 User tries to create an account with an empty username or password. Does the app check that both fields are valid before creating an account?
    - 11.1.3 User logs out of one account then logs in to another account. Does everything work as expected (nothing from the first user session appears in second user session)?
    - 11.1.4 User exits the app (without closing it), then re-enters. Does the app resume normally?
    - 11.1.5 User is signed in, and they close their app completely. Will they still be signed in when they reopen the app?
    - 11.1.6 Does the UI scale correctly on phones of different sizes (and possibly tablets that run iOS/Android)?

Fayetteville Public Library  
Volunteer Time-Tracking and Management System

- 11.1.7 What happens if a user clocks in an event and never clocks out? Will the timer run indefinitely or stop once a time limit has been reached?
- 12. Begin port to mobile applications
  - 12.1 Use iOS and Android emulators (or real phones) to test our app experience to ensure that there are no bugs.
- 13. Begin aesthetic design of front end and mobile apps
  - 13.1 Include aesthetically pleasing fonts/pictures/descriptions and good color schemes
- 14. Test mobile apps
  - 14.1 After aesthetic changes have been made, use iOS and Android emulators (or real phones) to test our app experience
- 15. Final tests
- 16. Final Polish
- 17. Documentation

#### 4.5 Schedule

Tasks	Dates
1. Visit library to understand the process needed to be currently taken by volunteers	11/01-11/05
2. Understand how the library currently organizes its systems	11/08-11/12
3. Discuss and design the high-level view and expected features of the application with sponsors	11/15-11/19
4. Learn Flutter and how to build applications with it.	1/18-1/21
5. Begin the implementation of front end, bare bone implementation	1/24-2/4
6. Begin the implementation of back end	2/07-2/18
7. Align existing library volunteer database with back-end implementation	2/21-2/25
8. Finish implementation of front end	2/28-3/04
9. Test back end of the application	2/28-3/04
10. Link together front end and back end containing database	3/07-3/11
11. Test system as a whole	3/14-3/16
12. Begin port to mobile applications	3/17-3/31
13. Begin aesthetic design of front end and mobile apps	4/1-4/8
14. Test mobile apps	4/11-4/15
15. Final tests	4/22-4/26



16. Final polish	4/27-4/30
17. Documentation	5/02-5/04

#### 4.6 Deliverables

- Design Document: Contains a listing of each major hardware and software component
- Database schema: The specification of records in our database implementation
- Database code: The Ubuntu PSQL code for our database implementation
- Web Application: The front end of our system for users and admins to interact with the system
- Android & iOS code: The Python Django and Flutter code used for the application itself
- Final Report

#### 5.0 Key Personnel

**Nathan Secrest** – Secrest is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed relevant courses such as Database Management Systems, Software Engineering, Programming Paradigms, and more. He is currently taking Information Retrieval. Secrest is experienced in C++, Java, MySQL, and JavaScript. He has also had an internship at EngwIT, a software development company located in Gravette, AR. Secrest will be responsible for backend development.

**Kagen Crouch** – Crouch is a senior Bachelor of Science in Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed relevant courses such as software engineering and programming paradigms. Crouch has gained extensive knowledge in Java, JavaScript, and Python programming languages through classes at the university. Crouch also contains knowledge of Node.JS, HTML, and CSS. This knowledge was obtained through a software engineering class where he was reasonable for the creation of the front end of a mock supplement marketplace like that of Amazon. Crouch will be reasonable for the creation and implementation of the front end of our web application and the various mobile applications that will be created.

**Steven Trinh** – Trinh is a senior Computer Science major at the University of Arkansas. He has completed relevant courses including Database Management Systems, Software Engineering, and Programming Paradigms. He is currently taking Mobile Programming which involves Android development programming projects. He is knowledgeable in C++, Java, JavaScript, Python, and SQL. Trinh will be responsible for Android application development which includes storing/retrieving user information from the database when the user interacts with the application and getting the UI to correctly and responsively display information whenever a user performs an action.

**Bradley Lithalangsy** – Lithalangsy is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed relevant courses such as Database Management and Software Engineering, and he is currently enrolled in Mobile Programming. Lithalangsy is knowledgeable in VB.NET, Java, JavaScript, Python, and SQL. He has experience creating a pricing website through his internship using VB.NET and JavaScript. Lithalangsy will be responsible for the cross platform mobile application development in Flutter and helping with the front-end development of the web application.

**Alexis Carter** – Carter is a junior/senior Computer Science/Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed relevant courses such as Database Management, Information Retrieval, and Software Engineering. She has used Python, Java, JavaScript, JFlex, and SQL. She has built a database from the backend and frontend and used SQL to manipulate databases. She has also designed several simple multi-page websites and their interfaces.

**Carlye Dennis** – Dennis is the current Community Engagement Manager at the Fayetteville public library. Dennis has received her Bachelor of English at the University of Arkansas in 2004, while also receiving her master's in Library Science and Administration at the University of North Texas in 2008. Carlye has performed several tasks for the public library such as, developmental assistant and manager of volunteer and outreach.

**Chris Moody** – Moody is the current Directory of IT/AV at the Fayetteville Public Library. Moody received his Bachelor of Education at the University of Arkansas. Moody also worked under the university for 13 years performing several tasks one of which being IT administrator for 4 years.

**Kent Watson** – Watson has worked under several roofs over the years. Some of these companies include Tyson Foods, University of Arkansas, RockFish and Metova Inc. While at these companies Watson has served as several positions from PC support all the way to executive technology strategy. Watson has a Bachelor of Science and a Master of Science from the University of Arkansas.

## **6.0 Facilities and Equipment**

No facilities will be needed for the completion of the project. The only equipment that will be required is various mobile devices, but these can be emulated from groups workstations at home.

## 7.0 References

- [1] Django, <https://www.djangoproject.com/foundation/>
- [2] Flutter, <https://flutter.dev/docs>
- [3] Material Design, <https://material.io/>
- [4] PostgreSQL, <https://www.postgresql.org/about/>
- [5] GivePulse, <https://www.givepulse.com/>
- [6] Ubuntu, <https://ubuntu.com/>