CE-4020-004

Team Omicron

Milestone 5 Cycle Report

Submitted February 7th, 2019

# Executive Summary

LogiSteps is a full stack application that is designed to collect, process, and display user fitness data in a seamless, self-powered construct. LogiSteps allows a user to pair their Bluetooth enabled smart sole with their mobile device and stream data to the cloud in a manner that is unobtrusive and relies very little on the user. Prior to beginning winter quarter, five milestones were set, with the previous 4 being completed on-time, in addition to this fifth milestone. The completion of this fifth milestone, and hence, fifth development cycle, symbolizes the functionality of the web application laid out in the system requirements. This development cycle brought about no major needs for redevelopment or adjustment of system plans.

# Introduction

The Logisteps project has reached it’s 9th week of development and has remained on schedule for a majority of milestones and sub-milestone items that were identified early in the winter trimester. The Logisteps smart insole technology was completed mid-way through the trimester, with the mobile application and web application either close-to, or at initial completion goals. The only Logisteps sub-system that is lagging slightly behind schedule is the microcontroller application, largely due to difficulties with the Sparkfun breakout board (which will be discusses later in this documents). All major milestones have so far been met, putting Logisteps in a good position to soon begin system integration. The newest major milestone that caps off the fifth development cycle is the completion of initial web application functionality, including both backend (web server, mobile API, database) and frontend web technologies.

The remainder of this document will go into further detail regarding risks to the project, as well as possible contingency plans that are ready to be implemented if needed, overall performance of the team regarding project plan, the deliverables created for this fifth development cycle, updates to development and planning, followed by a final conclusion. The completion of this fifth milestone marks a significant pivotal point in Logisteps progress.

# Discussion

The completion of the fifth development cycle caps off a major deliverable required for usability of the Logisteps system – the web application. While this is not the only progress that was made during this cycle, it is the only major deliverable that was planned for completion. During the completion of this fifth milestone, other progress has been made in other sub-components of the project, which will be discussed briefly as well.

## Web Application

Development of the web application began in the beginning of the winter trimester and has progressed to the point where it now meets the system requirements that were laid out at the beginning of the trimester. The web application was developed in two separate phases; one completing the backend services needed by the mobile API and database design, and one completing the frontend user-facing interface. For simplicity, these two phases will be discussed in their own sections.

### Backend Development

Backend development began in the first week of the winter trimester, and the major goals of this phase of the project was to get the infrastructure developed which would be required for core capabilities and the mobile application API. By completing this early in the design phase, freed up sub-components of Logisteps that were dependent on communication with the web server, such as the mobile application.

This development began slow, since the web framework being used (Django) was unfamiliar, and much of the database schema and API interfaces were not planned yet. To begin, a tutorial Django app was developed, with the purpose of learning the features of the framework, and work began developing the database schema. The database schema work proved to be essential, as it became the foundation of the Logisteps web application. Through several iterations and modifications, a finalized data model and consequently, a database schema was developed, which can be found in the design documentation for the web application.

After finishing the planned design for the Logisteps data model, work began implementing the object models in the Django web application. Implementing the models in Django was very straightforward and easy and didn’t involve any major roadblocks. After implementing the data models in Django, the models were used to seamlessly roll out the database schema. Completion of this task put the web application in a state where data could be stored and modeled, but there was not yet any interfaces for creating, manipulating, or deleting data.

The next effort relating to the web application was constructing an interface for the mobile application to create user and step data, as well as authenticating users and pulling down relevant model data. While the previously completed design efforts went relatively smoothly, this phase of the project quickly hit a temporary roadblock. It turned out that Django makes efforts to closely couple the backend web server to the frontend web browser interface. As a result, requests being made to the web server from an outside resource, such as the mobile app, were rejected. This turned out to be the default behavior of Django and getting around it is a tedious task. After doing some initial research, a third-party plugin for Django called “rest-framework” was found, which helped ease the burden of developing a rest framework designed to service clients other than a web browser. Using this plugin made it possible to begin design Django views to service requests coming from the mobile application.

Using the rest-framework plugin, web endpoints were designed to allow for user creation, user modification, user deletion, and step posting. These endpoints were designed and implemented first because they would satisfy all major requirements of the mobile application. For the most part, this phase of design and implementation went well. The only difficult aspects were learning how the Django user authentication system works, how to extend the user authentication system for the needs of Logisteps, and the process for serializing and deserialzing objects for web communication since Python does not natively work well with JSON. After completion of this, the first major phase of the web application was finished, and James able to begin using the web application to test send data from a user’s mobile phone to the web server. Figure 1 illustrates the progress made by this point in web application development.

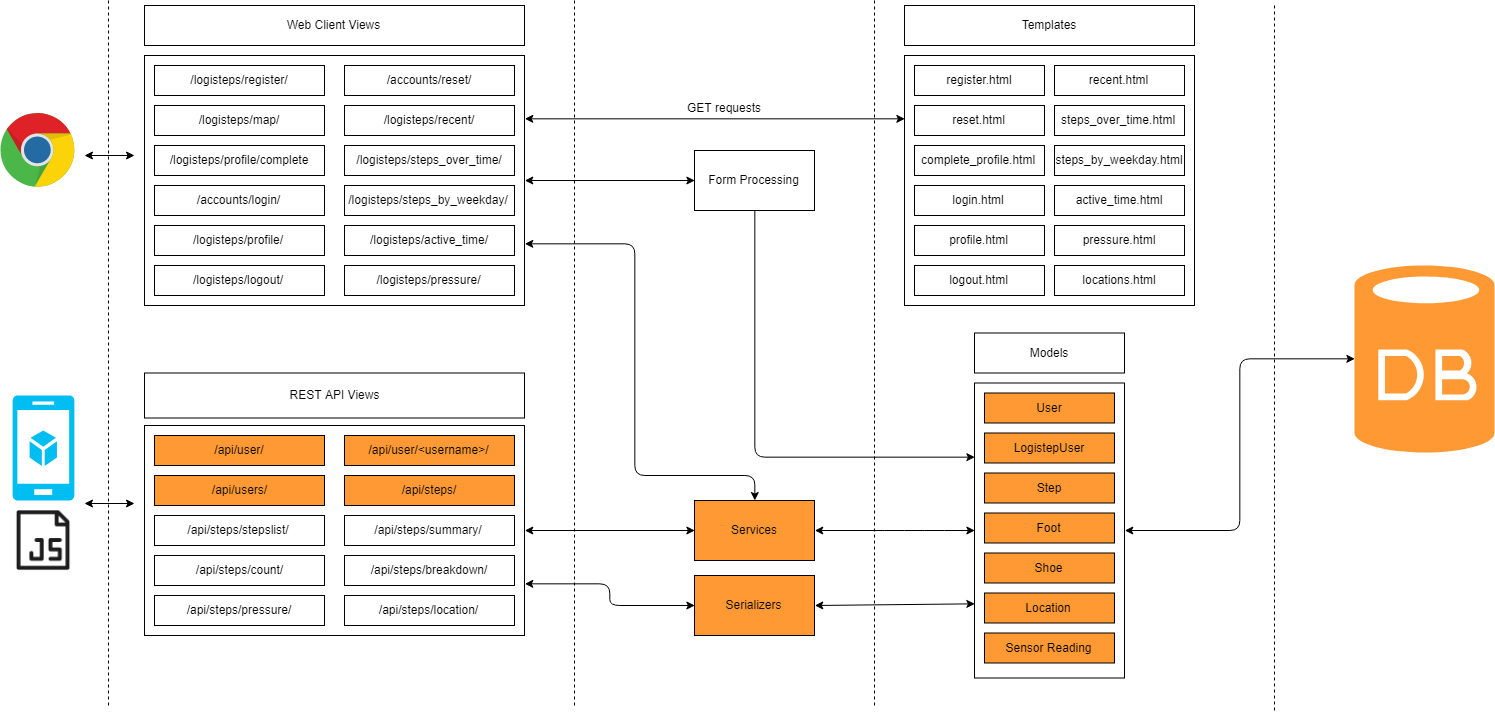


Figure - Highlighted in orange is the components of the web app completed in the backend phase of its development.

### Frontend Development

Completion of backend services and architecture meant that work could begin on the user facing side of the web application. This work began about halfway through the winter trimester and wrapped up in week 9 of the winter trimester. In brief, this work included user authentication views, user profile views, as well as all the views presenting rich graphics illustration a user’s step data.

The first phase of this development was targeted at completing the interface for registering users, logging in, and logging out. This work proved to be relatively straightforward, and the Django documentation did a good job documenting much of the process for doing this. This user authentication system blended into the mobile application authentication system well, and synchronized user data between the mobile app and the web interface. An example of the resulting user authentication system can be seen in figure 2 shown below.

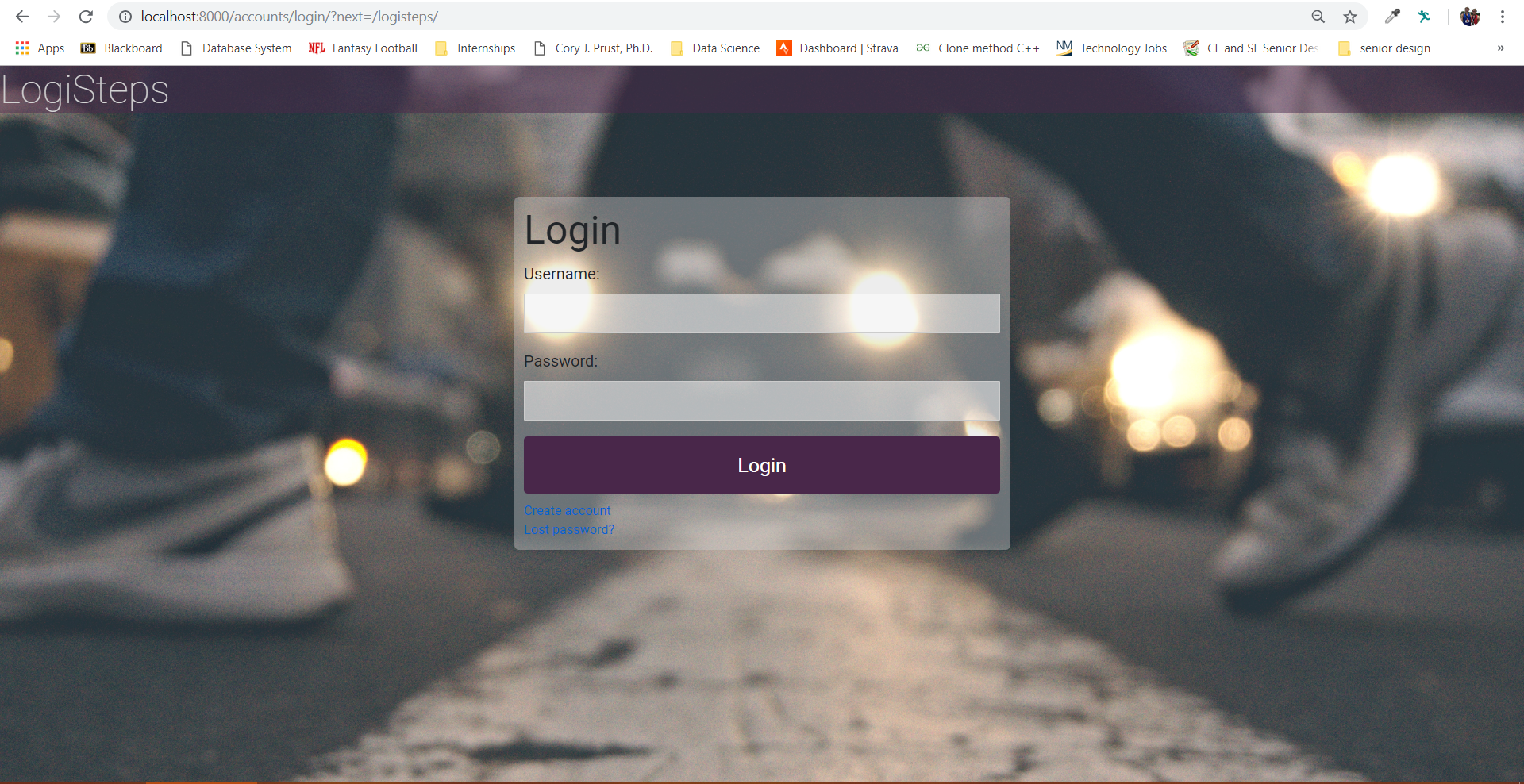


Figure - Screenshot of the login interface for a Logisteps user.

Following completion of the user-facing authentication system, work began constructing the main portion of the web application, which involved calculating analytics and assembling the web pages for presenting them. This portion of the development process was likely the most difficult, as it involved large calculations of subsets of the database and required the use of a third-party graphics library. Initial plans were to user D3JS for graphics, but a much more simplified library named C3JS was found which provided the views needed by the application.

Another difficult aspect of this portion of this sub-component was the need for test data. Without a significant amount of data loaded in the database, it was difficult to test the graphics libraries and analytical functions that were being used to display step data to users. This problem was tacked by using a data generator to generate 10,000 pseudorandom step data objects over the timespan of 1/1/17 to 12/31/18. This pseudorandom data was then used to verify correct functionality of both the analytical web server functions, and the rendering of the graphics.

User facing graphical views of Logisteps user data were created using custom database queries, custom Python data service functions (used for calculating analytics), Python view handlers, HTML templates, LESS (CSS tool), C3JS graphics libraries, and custom JavaScript used to load the data front end. A major hurdle that was encountered during this phase of the project involved the process of getting data to the C3JS graphics libraries. Django provides template shortcuts for displaying dynamic data in web pages, but it lacks the ability to send this data to frontend JavaScript files, which is where the C3JS function calls are made. To solve this, extra endpoints were added to the exposed rest-framework, and XHR web requests were made from the frontend JavaScript files to gather Logisteps analytical data and load it into the graphics libraries.

Final completion of the user-facing web interface involved the creation of dozens of new files and resulted in web views like the example shown in figure 3.

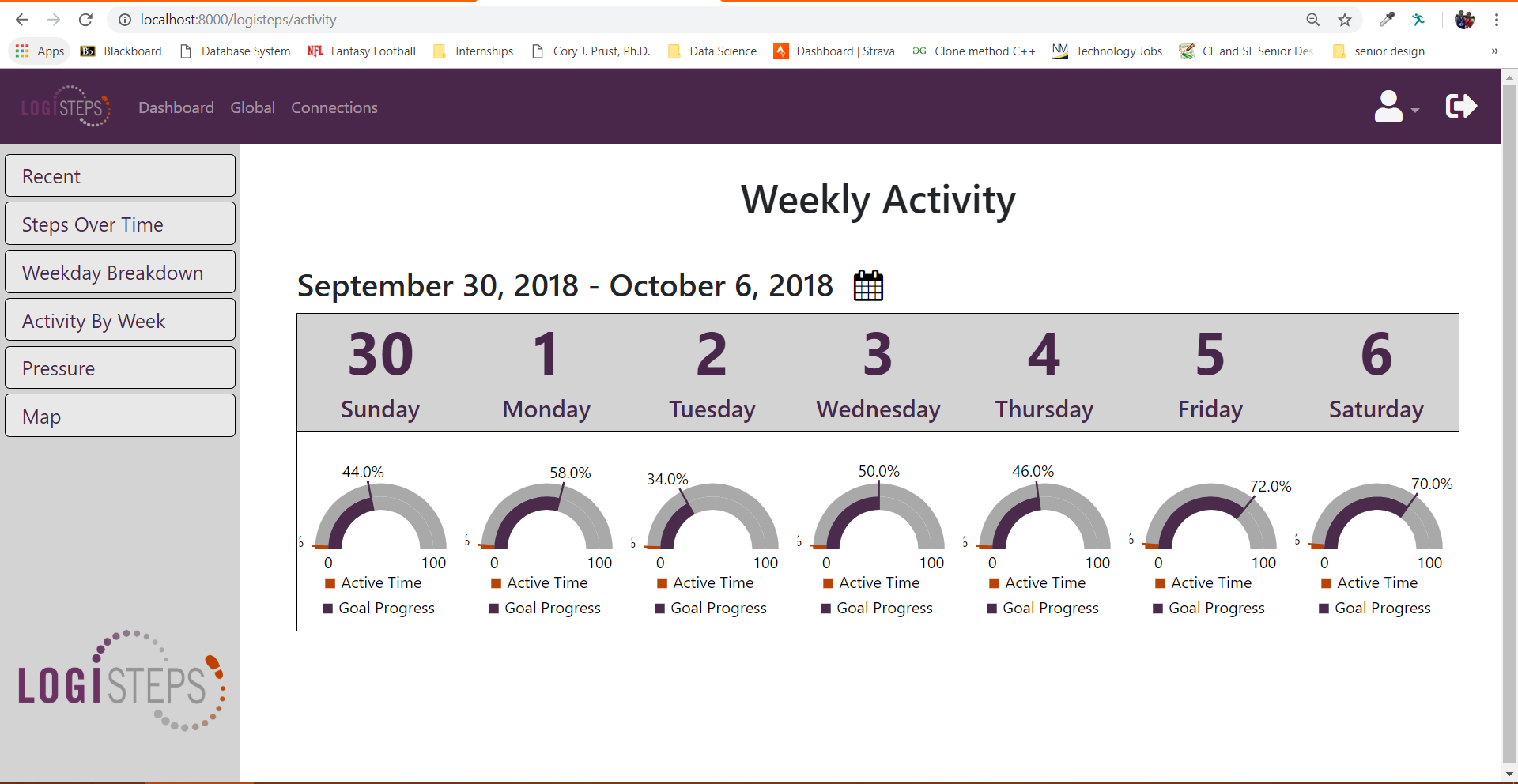


Figure - Shown here is a screenshot of the Weekly Activity view, which presents a user with a view of their goal completion and active time vs. inactive time for each day of a given week. The week can be changed using the calendar icon, and new data is dynamically loaded into the view.

Upon completion of the frontend development of the web application, all system requirements that were previously defined in fall of trimester 1 were concluded to be met.

# Risks to the project

During development of milestone 5, risks to the project related to the web application were reduced to a minimum, as all functionality required by the Logisteps project has been implemented. Most of the work that is left for the web application is cosmetic visual improvements. While risk has significantly decreased in this sub-component, risk to the project has increased in the microcontroller sub-component. This is work that is not related to the web application, but during the fifth development cycle, while beginning integration of the microcontroller with the insole, power problems became apparent. The issue that has risen is the Sparkfun breakout board is causing a high-power consumption than what was originally planned and documented in the Nordic datasheets. This is caused by Sparkfun’s failure to expose the pin required for DCDC power. Power consumption must be brought down to a low enough level so that it can be passively powered by the insole.

To help migrate the risk that this problem poses to the Logisteps project, several different contingency plans have been developed, with the following priority.

1. Remove the power regulator on the Sparkfun board. There is speculation, based on datasheet information and discussion on internet forums that desoldering the power regulator on the Sparkfun board can help reduce power consumption to the levels required by our design (microamp current range).
2. Purchase a new breakout board that exposes the pin needed for Nordic DCDC operation. Since all the embedded software is portable, doing this would be a relatively easy process.
3. In a last-ditch effort, a supplementary battery could be embedded into the sole. This is a last resort option and is being avoided as much as possible.

# Deliverables

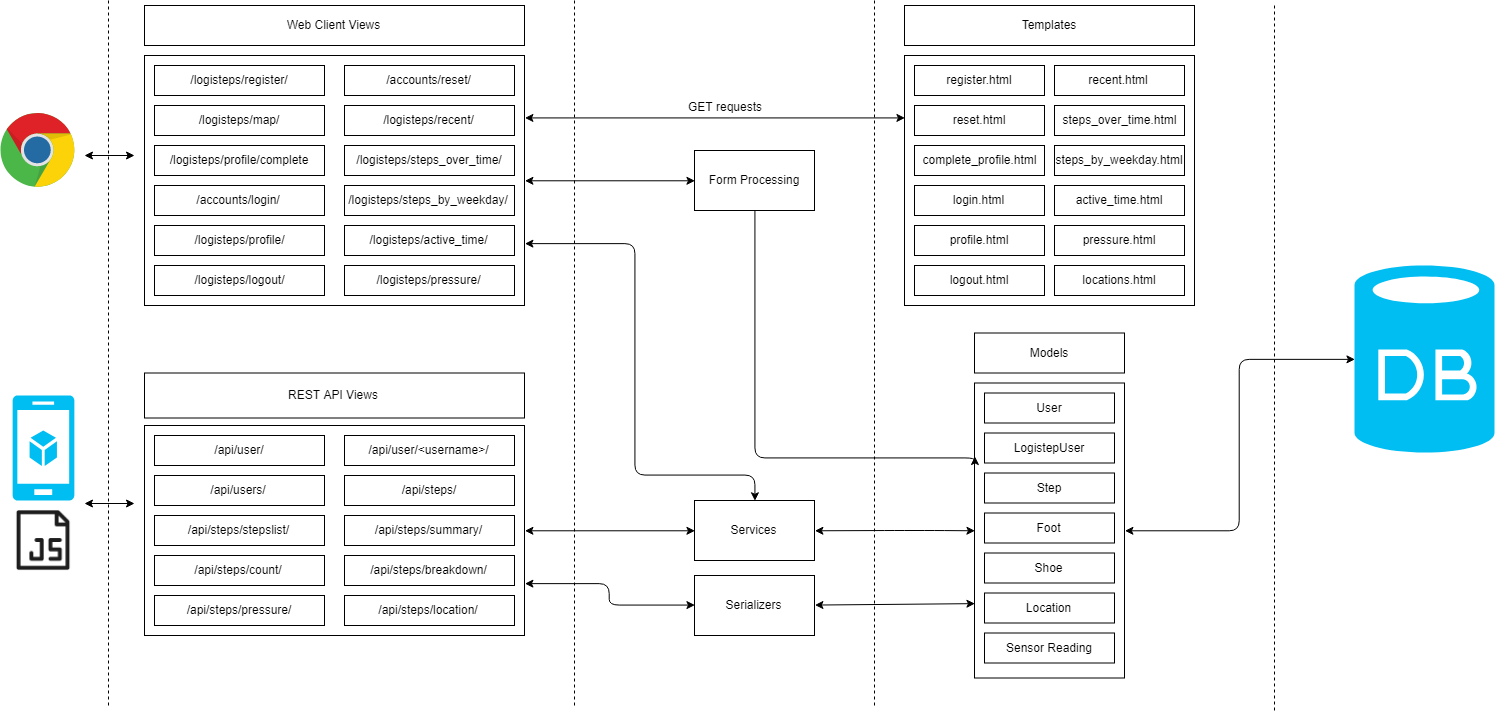
The following list summarizes the deliverables completed for milestone 5. This is a summarized list of all deliverables and not an exhaustive list of every single task completed.

* Web Application
  + Backend
    - Model design and implementation
      * LogistepsUser, Step, Location, Shoe, SensorReading
    - Serializers
    - REST API endpoints
    - Permissions
    - Database
    - Authentication system
    - Data services
  + Frontend
    - HTML Pages
    - Authentication system
    - User forms
    - View handlers
    - Styling sheets (CSS, LESS)
    - JavaScript files
    - Graphics
* In addition, a test plan has been developed to test functionality of the web server

In total, nearly 3000 lines of code were written for milestone 5, through 72 commits to github.

# Development Plan / Plan Update

Implementation of the web application did not deviate much from initial design documentation, as a result, only a few development plans were changed. In particular, the mobile application endpoint which returns a breakdown of user’s pressure data, aggregated over the course of a day, week, and month had its response object schema changed. In initial documentation, the endpoint would return an object only 3 values – average pressure for day, week, and month. This was changed to support multiple sensors, for each foot. As a result, the endpoint now returns the average pressure placed on each sensor, in each foot, averaged over a day, week, and month. Final documentation can be found at the [SeniorDesignTeamOmicron](https://github.com/SeniorDesignTeamOmicron)/[WebApplication](https://github.com/SeniorDesignTeamOmicron/WebApplication) github repository. A summary of final documentation is attached in the following figures.



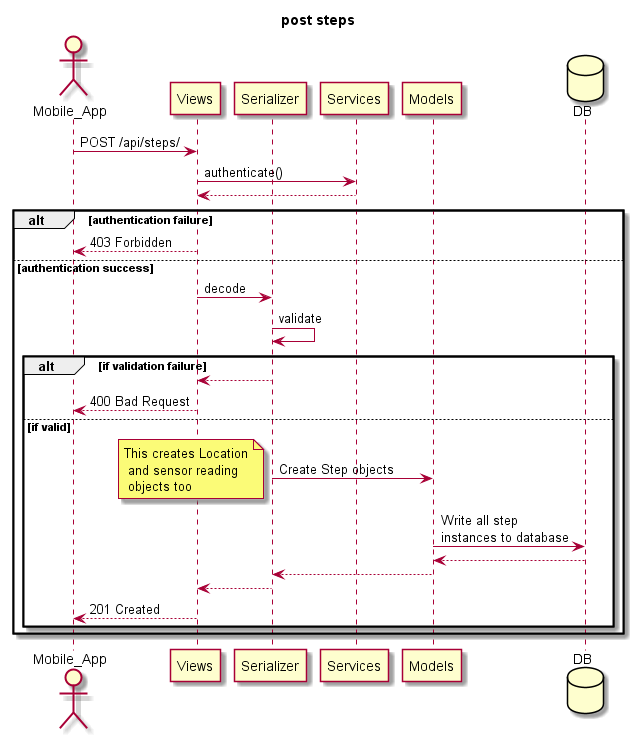
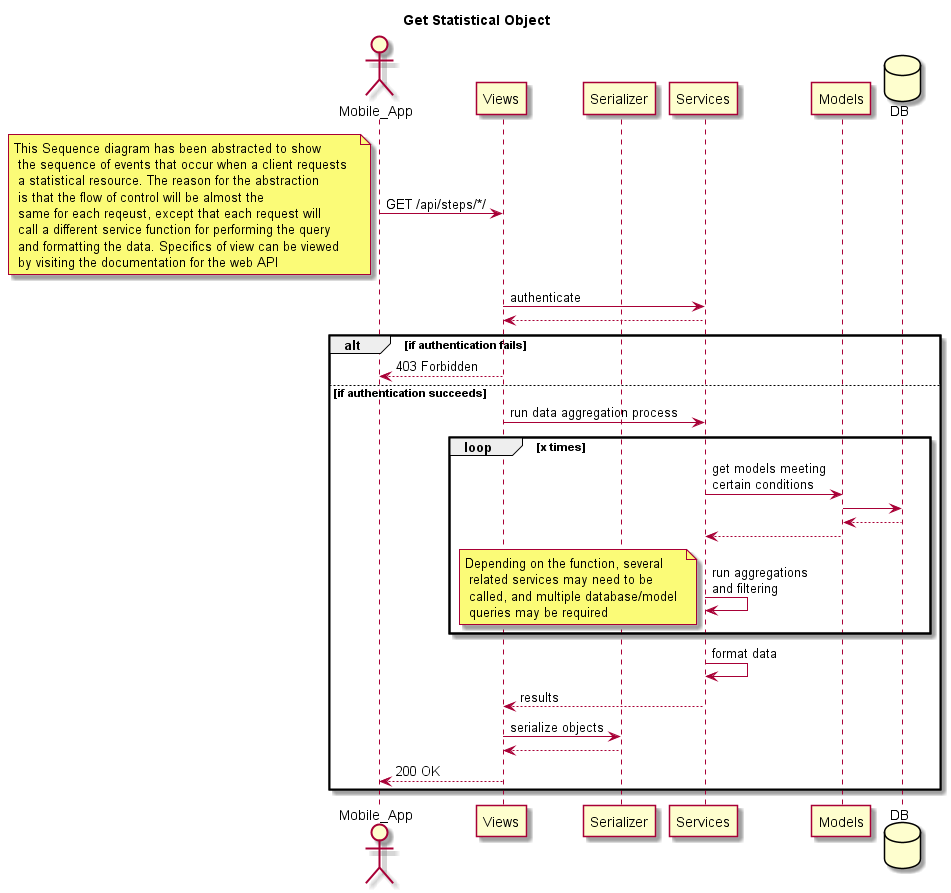
Figure - Full architecture of the web app.

Figure - Get statistical object.

Figure - Posting step data to web server.

Planned deliverables for the next development cycle are integration of sub-components, but this development cycle is still being planned.

# Conclusion

The completion of the fifth milestone, and fifth development cycle, brings about a finish to individual implementation of sub-components. Moving forward, Logisteps plans to shift focus to integration of components, and testing of the system. Successful integration of system sub-components will prove to be a major accomplishment for the Logisteps project.