



Camosun College Capstone Project

Camosun Innovates Nursing Survey

Final Report

Prepared for
Katie Tanigawa
Jonas Bambi
Rich Burman

Prepared by
Mike Vandendorpel
Aleks Janzen
Meron Meresa

Last Updated:	2022-08-07
Document Version:	1.1.2

Memorandum

To: Katie Tanigawa, Jonas Bambi

From: Team J

Date: August 7th, 2022

Re: Final Report for the Camosun Innovates Nursing Survey

Dear Katie and Jonas:

The following document is our final report for the Nursing Survey project, a web application designed request of Rich Burman from Camosun Innovates. This report will give a brief overview of the development of the project, as well as give an idea of where the project currently stands, and ideas on how to further develop the web application.

Although we have created a working proof of concept for the web application, many challenges that were not foreseen took up a lot of the development time. These challenges led to some excellent learning opportunities for the team, but also meant that we were not able to deliver on all of the features requested by the project sponsor. In the end, we were still able to deliver on the largest of the requirements of the project: integrating the public Fitbit API into a secure database that can export the data into a useable format for the researchers, and an intuitive frontend interface for users.

Any questions or concerns regarding this report, or the project itself, can be directed to Mike Vandendorpel at chaosengel@gmail.com.

Thank you for your interest in reading this document.

Sincerely,

Team J:
Mike Vandendorpel
Meron Meresa
Aleks Janzen

Executive Summary

The Capstone Project for this team was to develop a web application that would gather data from user surveys and Fitbit devices and export this data into spreadsheet format for analysis. The team decided to break this project up into four smaller projects:

1. A front-end website
2. A middleware API
3. A relational database
4. A separate tool for gathering exported data

This project was developed using Agile methodology, and sprints were two weeks long. All tasks were broken down and assigned to the team members based on which section they were maintaining.

Although the team encountered many challenges in the design and implementation of this project, we were able to deliver a proof-of-concept application that can be further developed by another team in the future.

Top priorities for this project were to make the front-end intuitive and simple for users, gathering data from Fitbit's API, securing user data at-rest and in-transit, and producing documentation to ease passing off the project to another team.

The team had an opportunity to put many of the concepts we have learned over the past two years into practice, and although the challenges faced took up much of the development time, many of these skills are not applicable solely in web development but can be built upon in future careers in other disciplines as well.

Most of the project will be delivered to the Project Sponsor through a GitHub repository, although the more sensitive data, such as the Fitbit Developer account credentials, will be handed off privately.

Because the team was only able to develop a proof of concept for this project, there are many more steps that could be taken to complete this project by another team. A few issues that have not been resolved, either because of the limitations of developing a webserver inside a VPN, or because the team did not give itself enough time to properly implement a feature of the web application.

Table of Contents

Project Description.....	1
Project Goals and Objectives	1
List of Deliverables.....	2
Development Environment	2
Implementation Environment.....	2
Project Metrics and Resources	3
Successes and Lessons Learned	4
Tying It All Together	5
Follow-Up Requirements or Recommendations	6
References List	7

Project Description

This project involved the creation of a web application to be used in a study of stress and burnout related to perioperative nurses. The web application would gather data from users completing surveys and from collecting biometric data from a user's Fitbit device. This data would then be exported to spreadsheet format for review and analysis by the research team.

Other requirements were that the data had to be secured at rest, that the web application feel intuitive and easy-to-use, and that the web application would be viewable from a mobile device, such as a smartphone or tablet.

The team decided to develop the application in a Node.JS environment, and that a relational database, such as MySQL would be preferable over a NoSQL database. React.JS was used on a front-end, along with modules that supported a "mobile-first" development strategy.

As an API would be necessary to regulate communication from the front-end and the database, as well as allowing communication with Fitbit's own API, Express.JS was used to build middleware for the application.

Project Goals and Objectives

The goals for this project were as follows:

- Design and develop a secure website that gathered data from users through surveys conducted daily and weekly.
- Design an encrypted database to store users' personal information securely.
- Complete documentation to allow for transition of project to another team.
- Develop a report generation tool that exports data to Excel.
- Develop an API to gather data from Fitbit devices and handle all communication between the frontend application and the database.

List of Deliverables

All content created for this project will be delivered through a public GitHub project, including the following components:

1. All source code with comments for future developers for:
 - a. The front-end
 - b. The API
 - c. The research generation tool
2. An SQL script for importing the database to a new server
3. A readme.md on the GitHub repository to give an overview of the project
4. Any other needed documentation for the project(user guide, database modelling, etc.)

The Fitbit Developer and Gmail accounts and passwords will be handed off separately to Rich Burman personally.

Development Environment

The project was developed on a Virtual Machine provided by Camosun for use by the team. The VM was running Ubuntu Server, and the team developed the front-end and API through Visual Studio Code connecting remotely. The MySQL server was accessed using MySQL Workbench.

The VM was configured to run Node.JS and hosted a MySQL server used for testing the project. Several modules were installed to support development of the website and API.

Implementation Environment

The API can be run as a server in a Node.JS environment. The team designed the frontend to run in this same manner or could be built as an application for users to run natively on their mobile device or PC.

The database can be run on a MySQL server and should be easily convertible into an Oracle SQL database if desired, although any security configurations on either setup will have to be done manually.

The “process.env” file stores any needed connection information for both the database and the Fitbit public API. Although the Fitbit data will be removed from the file on the GitHub repository, this data will be given over to Rich Burman, who may change access to these accounts as he sees fit.

Project Metrics and Resources

This team designed this project using Agile methodology and each task was broken down and assigned an arbitrary unit of measurement regarding time and difficulty of the task. The team referred to these units as marmots for our own amusement. Sprints were two weeks long, as required by the Capstone class. Tasks were assigned each sprint to the team members, although, each team member was assigned to a specific part(frontend, API, database) of the application.

These units were calculated at the end of each week, and were recorded on the project's burn-up chart at the end of each sprint, show in Figures 1 and 2 below:

Sprint	Project Total Units	Estimated Units This Sprint	Completed This Sprint	Cumulative Completed	Estimated Trajectory
0	178	0	0	0	0
1	178	35	23	23	35
2	218	35	56	79	70
3	240	35	64	143	105
4	240	40	21	164	145
5	248	40	48	212	185
6	293	40	83	295	225

Figure 1- Tracking of marmots completed each sprint[1]

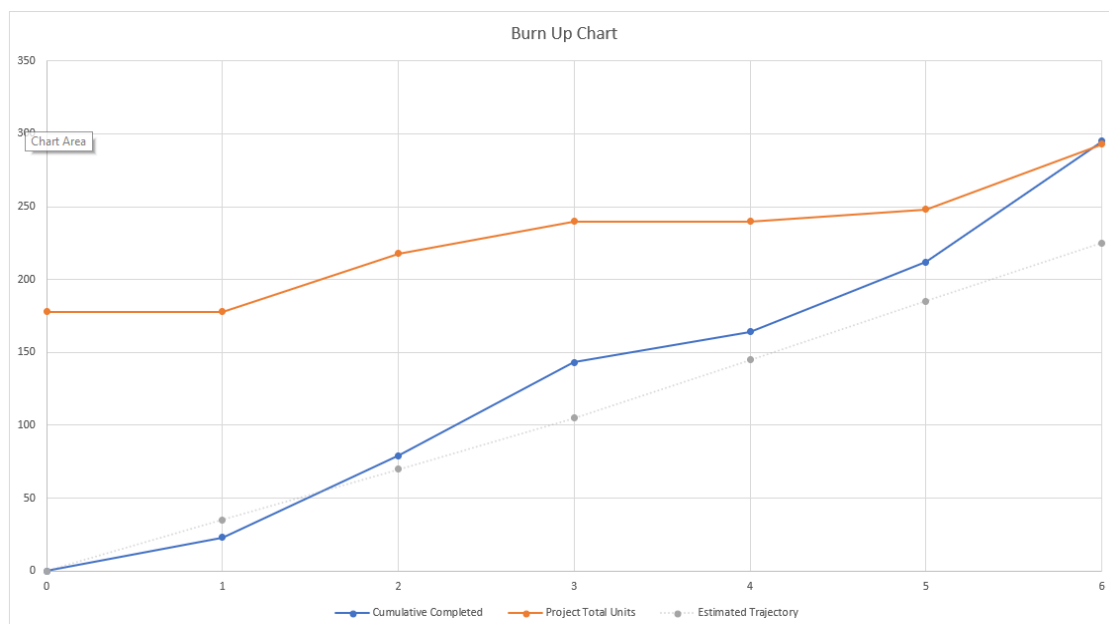


Figure 2 - Project Burn-up Chart[1]

Resources for this project were limited to our own research, and what information and experience our instructors had. Because major parts of this project required integration with a specific API, Fitbit's documentation was a major resource for developing the API, as well as the forums Fitbit provided to developers.

This project was designed to be no-cost, and all technologies used were open-source and relied on their own documentation for this most part. This included their GitHub repo pages, and

documentation for modules on the Node.JS website, as well as often looking through discussions revolving completely unrelated modules. Google search was certainly the largest resource used by the team.

Successes and Lessons Learned

Although some parts of the project were not completed, or not completed properly, the website is functional, and gathers data from both the user surveys and Fitbit's API. This data is stored encrypted into the database, and the API is configured to decrypt this data before transmitting to the frontend application or exporting the data to spreadsheet format.

The team acquired many skills in full-stack web development, securing data-at-rest and data-in-transit, and developing a project in an Agile manner with all the accompanying paperwork. There were many tough lessons learned on how to operate as a team working on a single project, often working at the exact same time. There were also opportunities to learn how to overcome obstacles and bugs as a team working together on different parts of the project in tandem.

Mike Vandendorpel

I personally gained quite a bit of experience in OAuth2 authentication, and many of the potential bugs that can occur, as well as integrating with another public API. I also spent a lot of time learning how to manipulate and calculate data and format it from multi-dimensional JSON data into two-dimensional spreadsheets. I learned how to setup a development webserver from scratch, including setting up a database and development environment for a team.

Aleks Janzen

My biggest personal success with building this project is being able to put the knowledge that I had gained during my time at Camosun to a large-scale project that will be put to real-world use. Working on the project helped me cement that knowledge and the role that I would like to be working in when I make it further into the tech industry. As for the lessons that I learned throughout the project, the key point for myself was the importance for each of us to be communicating together with what we're working on or what we may be having issues with. Additionally, I learned on how important it is for a project like this to have a supportive sponsor who gives clear guidelines.

Meron Meresa

Some of the successes I encountered during the project involved the design of the database. One of those successes was implementing the table structure and ensuring both myself and the database remained flexible when data needed to be changed or added. Another major success was encrypting the database. It required a lot of Google searches and scrolling through Stack Overflow. A major lesson I learned was how valuable git branches are when working on a shared project. I once made the mistake of working on the main branch and caused a few features to break in the process, which could have been avoided by making a new branch.

Tying It All Together

What technologies/concepts that you learned in the program were useful to your project?

Many of the concepts taught in our second-year web development courses(ICS 211/221) were instrumental in developing this project. Parts of the API's code are taken directly from labs completed in ICS 221.

Learning how to develop in a Node.JS environment, setting up a VM and server, database management and administration were all concepts learned in that last two years that we have had a chance to put into practice over the course of this project.

What technologies/concepts did you learn for your project that was aided by what you learned in the program? i.e., java programming and jsp aided in learning java server faces

Although not explicitly covered some of the previous instructors' instructions and exercises on navigating documentation in general made navigating through Node.JS documentation was made much easier by. While there is no actual class on "How-To Google for Programmers," it is a concept that our team has been able to practice often in the last two years.

What technologies/concepts should we have taught in the program? i.e.: web technologies, source control systems etc. (not specific languages)

Oauth2 is a standard method of connecting with other systems in web development, and although we were given an overview of the concept, and how to implement it in its most basic form, a more structured example of connecting to a major Oauth2 authority, Google, for example, could be useful for students and is a hurdle that has come up in both Capstone, and ICS 199 by some teams.

Follow-Up Requirements or Recommendations

The API and front-end both require SSL certificates to be operable over an HTTPS connection which the Fitbit API requires. Both are currently configured with self-signed certificates to run in a testing environment but should be configured with proper certificates from a Certification Authority.

The daily survey part of the front end is currently not properly implemented. The daily survey first asks if the user is working a night or day shift, then sorts the questions based on that answer. The original intention was to have two daily surveys, one completed in the morning, and another completed at night. The survey questions are stored in the database and have been divided into these two surveys. Also, if the user selects that they are working the night shift, there is no chance for the user to enter in what shift they worked.

The JWT passport implementation has not been fully tested and implemented. Currently, data the API requires is passed in through query strings, which is not secure. The query parameters should instead be encoded into the JWT token and decoded by the API. Implementing this, and then passing the token through the Fitbit API's state parameter, will help prevent XSRF attacks on the API.

The research tool is an HTML page that performs an API call to the research endpoints on the API. This tool could be expanded upon, although using another programming language such as Python would be a better idea. Some tests were done that allow the data gathered from the downloaded spreadsheet to be exported dynamically to graphs, but we did not give ourselves enough time to build a proper tool to handle this.

Fitbit provides OAuth2 authorization tokens, and this could be implemented on the frontend website to remove the need for users to have a separate account with the web application itself, as Fitbit is already an integral part of the survey, users would already need to have an account, and could log into the web application using these credentials.

References List

- [1] M. Vandendorpel and A. Janzen and M. Meresa, '*Weekly Status Report #13*', Victoria, 2022.