



# Lean Analytics for a Leaner Person

## Demo #2 – Team Contribution Breakdown

## Fall 2015, Group #8 – Information & Members

Project Repository @ Github – [Health Analytics](#)

Project journal – <http://blog.willkara.com>

Website Homepage – <http://willkara.com/projects/HealthAnalytics/index.html>

Manuel Maldonado – [manuel.maldonado@rutgers.edu](mailto:manuel.maldonado@rutgers.edu)

Ming Tai Ha – [ming.tai.ha@rutgers.edu](mailto:ming.tai.ha@rutgers.edu)

Nick Taylor – [nitaylor@rci.rutgers.edu](mailto:nitaylor@rci.rutgers.edu)

Tongpeng Zhang – [tongpeng.zhang@rutgers.edu](mailto:tongpeng.zhang@rutgers.edu)

William Karavites – [willkara@rutgers.edu](mailto:willkara@rutgers.edu)

## Contribution Notes:

Listed below are the contributions each member was responsible for in demo #2. We broke down the contributions per team member and followed the items presented in the [grading format](#) posted on the professor's website.

For this demo we presented a working state-level health model (an improvement from demo #1 since we only had a national-level model). Since our system depends on users generating data, via signing up and using our website and mobile application, Dr. Marsic gave us feedback about data generation to show that our model works and to have a state-level model instead of a national-level one. Improvements on the model was the bulk of the work for this second iteration, finding the right data sources with the right dimensions, filtering, cleaning, formatting and utilizing the methods explained in the presentation to generate and validate a total of around 3.3 million users, or around 1% of each state (more details on how we did this will be given in the report). In the meantime, some of our members set-out to finish the integration of our full system as we decided that such heavy improvements in data generation (although great and definitely powerful) might not be enough to show how the system works as a whole. We were able to build the *working* prototype of the system, fully-integrated and ready for a small batch of initial users.

We completed development on all of our systems, albeit we implemented a substantial portion of our requirements (around two thirds of our full list), we chose to focus on the core use cases, foregoing "nice-to-have" features like social medial integration. We will explain the extent of implementation and decisions in our final report.

As far as integration goes, we completed the integration of our Health Model module (which runs once a week, although we were running it manually for the purpose of our current system) and the Datastore module. Also, we completed integration between the Database system (MySQL), the Datastore module and the Processor module which enabled the entire back-end functionality of our system. Finally, we fully integrated our user interfaces (mobile application and web application) and our back-end system leveraging our RESTful-like API service which was implemented by one of our members (link to the help file [here](#)). It should be important to note that since we built our system in a modular way and it is not tightly coupled it was advantageous for us as we could work separately as long as we followed our integration guidelines/requirements/documentation.

There was unit testing done on the health model and data generated, but it was a manual effort, as seen in the presentation. The Processor and Datastore modules included unit test cases (in code) for all methods with a 100% coverage of all the high-level methods. The mobile application and website also have unit test cases (also written in code) for all controllers and services with 100% coverage of those. All of these were not shown in the demo (because time constraints) but will be shown in the source code file archive with instructions on set-up and running them.

Data collection in our project is two-fold, but should be explained so the contributions are understood. The data used for creating our state-level model and generating our 3.3 million users was from NHANES, NCHS and Census 2010 (links to them and how to pull them will be shown in the project archive along with steps to fetch the data, etc.); all of this data was found after researching and had to be fetched, extracted, cleaned, filtered and ingested. Our second source of data is our system which

allows users to interact with it (via mobile app and website) and they will enter data using the interfaces. This second method of data collection is part of the system's functionality and requirements and it *is* supported by our system as of this demo.

Finally, our team is broken down into sub-groups (or components), one area per module, with a main person responsible and other members for support in specific areas, the table below shows this responsibility. The weight of each component represents the percentage of work done for demo #2.

*Table 1 - Sub-group Breakdown*

<b>Sub-Group (Component)</b>	<b>Weight (%)</b>	<b>Responsible Team (first person is the lead of that sub-group)</b>
<b>Mobile Application</b>	10%	Manuel
<b>Website</b>	10%	William
<b>Processor</b>	10%	Nick, Tongpeng
<b>Datastore &amp; Database</b>	10%	Nick
<b>User Generation</b>	30%	Ming, Manuel
<b>Health Model</b>	10%	Ming, Manuel
<b>Documentation</b>	10%	Everyone
<b>Project Management</b>	10%	Manuel, Ming

*Table 2 – Contribution breakdown in percentages for demo*

<b>Component</b>	<b>Manuel</b>	<b>Ming</b>	<b>Nick</b>	<b>Tong</b>	<b>William</b>
<b>Mobile Application</b>	100%				
<b>Website</b>					100%
<b>Processor</b>			95%	5%	
<b>Datastore &amp; Database</b>			100%		
<b>User Generation</b>	30%	70%			
<b>Health Model</b>	10%	90%			
<b>Documentation</b>	20%	20%	20%	20%	20%
<b>Project Management</b>	50%	35%			15%
<b>Total work for Demo #2 (normalized)</b>	27.00%	35.50%	21.50%	2.50%	13.50%

## Contributions by Member:

- **Manuel Maldonado:**
  - Program Code:
    - Mobile Application – Implemented full functionality.
    - User Generation – Implemented scripts for data collection.
  - Unit Testing:
    - Mobile Application – Unit testing using 'Karma' of all AngularJS controllers and services.
    - Mobile Application – Manual testing of interface and usability.
  - Webpage/Mobile Design:
    - Mobile Application Design – Implemented and design all.
  - Integration Testing:
    - Mobile Application – Karma-driven and manual integration testing between mobile application and Processor.
  - Documentation:
    - Mobile Application – Wrote README file with instructions on how to set-up, build, test and run the Mobile Application.
    - User Generation – Wrote README file on how to use all data fetching, cleaning and formatting used in user generation.
  - Data Collection:
    - Census 2010 – Researched and implemented scripts that programmatically downloads, filters and formats this data.
    - CDC/NHANES Data – Implemented scripts that extracted this data into the format needed for generation.
    - NCHS – This data was manually downloaded (PDF file), transcribed and formatted.
  - Brochure:
    - Updated Brochure to reflect current changes of our system.
  - Slides Preparation:
    - Aggregated all slides into the presentation seen.
    - Wrote information for the Mobile Application.
    - Demoed Mobile Application.
  - Project Management:
    - Organized meetings.
    - Helped with report formatting.
    - Helped coordinating activities.
    - Helped combining contributions into the correct files, formats and submit them.
- **Ming Tai Ha:**
  - Program Code:

- User Generation – Implemented the stratified Monte Carlo Sampling algorithm using python.
  - User Generation – Implemented the stratified Monte Carlo Sampling algorithm using python.
  - Health Model – Designed and chose final approach (Regression Tree) for our predictive Health Model.
  - Health Model – Implemented the regression tree algorithm for each state (seen in the demo) using python.
  - Health Model – State level user statistical attribute aggregation.
- Testing:
  - User Generation – Compared generated results with state-level and national level statistics (sources in presentation and will be added to the report).
  - Health Model – Tested for model by randomly split user generated data in half before generating the model. Half was used for generation, the other half for validation.
- Documentation:
  - User Generation – Wrote README files with instructions on how to configure the environment, install dependencies, and run generation and validation.
  - Health Model – Wrote README files with instructions on how to configure the environment, install dependencies, and run generation and validation.
- Data Collection:
  - Census 2010 Data – Implemented data ingestion for user generation.
  - CDC/NHANES Data – Implemented data ingestion for user and model generation, and validation.
  - Health Model Data – Implemented algorithm to export the Health Model in JSON format to a flat file that will be used by the Datastore to store it into our database.
- Slides Preparation:
  - Provided content for slides (specifically for Health Model and User Generation).
- Project Management:
  - Helped organize meetings.
  - Helped coordinating activities.
  - Helped with keeping team on track.
- Other:
  - Scientific research, extending our domain specific knowledge that aids the user-generation, validation and model generation.
- **Nick Taylor:**
  - Program Code:
    - Datastore – Implemented full functionality to support all API transactions and Health Model.
    - Processor – Implemented full functionality (list of all of the APIs implemented can be found [here](#), 18 total).

- Unit Testing:
    - Datastore – Unit tested (via code) all methods with 100% coverage.
    - Processor – Unit tested (via code) all APIs with 100% coverage.
  - Integration Testing:
    - Datastore – Fully integrated the Database with the Datastore via a PDO abstraction layer and with the health model via flat files.
    - Processor – Fully integrated the Processor and the Datastore. Provided integration documentation for use by the ones responsible of the website and mobile application.
  - Database:
    - Database – Designed and created tables for all data stored by our system. The database was created and maintained by Nick as well.
  - Data Collection:
    - Health Model – Read in health model files (6 files per state a total of 40MB of data).
  - Brochure:
    - Provided contents for Datastore and Processor.
  - Slides Preparation:
    - Provided content for slides (specifically for the Datastore and Processor).
- **Tongpeng Zhang:**
    - Program Code:
      - Processor – Helped implanting some functionality but it was not usable for the final version of our module.
    - Documentation:
      - System Architecture – Documented how our system architecture looks like and how it integrates with each module.
    - Brochure:
      - Provided information about the system architecture.
    - Slides Preparation:
      - Provided information and diagram about the system architecture.
  - **William Karavites:**
    - Program Code:
      - Website Design – Designed and fully implemented website views.
    - Unit Testing:
      - Website – Unit testing using of some controllers and services.
      - Website – manual testing of website using three major browsers (Chrome, Firefox and Safari).
    - Webpage/Mobile Design:
      - Mobile Application Design – Implemented and design all.
    - Integration Testing:

- Website – Code and manual integration testing between website and Processor.
- Documentation:
  - Website – Wrote README file with instructions on how to set-up, build, test and run it.
- Brochure:
  - Final formatting and updates before print.
- Slides Preparation:
  - Aggregated all slides into the presentation seen.
  - Wrote information for the Mobile Application.
  - Demoed Mobile Application.
- Project Management:
  - Helped with report formatting.
  - Helps combining contributions into the correct files, formats and submit them.
  - Runs the team's blog/website.
  - Owns the team's GitHub repository
  - Created the team's OneDrive shared storage.