

Deliverable 2

Design Document

Version 1.0 approved

Prepared by Group Nerd Herd

10/10/2017

1 Requirements Specifications	3
1.1 Scope	3
1.2 Functional Requirements	3
1.3 Nonfunctional Requirements	3
1.4 Interfaces	4
1.4.1 User Interface:	4
1.4.2 Software Interface	4
2 UML Design	5
2.1 Classes	5
2.2 Sequence	6
2.3 Use Cases	7
3 Test Plan	8
4 Project Plan (Update)	8
5 Risk Management (Update)	8
5.1 Updated Risks	8
5.2 Potential solutions	9
6 Meeting Minutes	9
7 Project Status	10
8 Member Contribution	10

1 Requirements Specifications

This document contains the defined requirements for the system we plan on implementing.

1.1 Scope

The purpose of this project is to provide an online web-service that provides the current estimated capacity of the UNT Pohl Recreation Center and its distribution of capacity amongst each of the gym's provided services (weight room, basketball courts, running track).

Our project has a finite amount of desired functions and capabilities. The scope of our project defines the boundaries for features and requirements that will be implemented into the system. These boundaries can be defined by the list of functional and nonfunctional requirements found in the following two sections. If a feature or operation is not found in those two sections, it is outside of the scope of our project.

1.2 Functional Requirements

This list of functional requirements defines what

- During certain times of the day, the application must pull and display the correct model to reflect the capacity.
- The application should pull a new model and update at interval times in real time (15 minutes, 30 minutes, 1 hour).
 - Between intervals, the software should take note when the swipe occurred and calculate a rough time estimate when that person will leave. Then subtract that person when the time spent aligns with the interval time for update. This might require a database in order to keep track.
- Alerts will highlight certain sections telling users that section is full; along with an estimated time of when that section will loosen up.
 - Each time a model is about to be uploaded to the application, data from that model will run through a function to see if it is close to a predefined capacity number and depending on the result, return an alert.
 - Using the time estimate database, find the total interval time in which the number of "supposed" leaving customers knocks down the capacity number to a non-full number and upload that time.
- Using a distribution model, the application must correctly distribute the total population into the sections of the gym.
 - Pool
 - Weight room
 - Courts
 - Yoga rooms

- Track
- Rock climbing
- etc...

1.3 Nonfunctional Requirements

- Responsiveness - In terms of actual speed the application must be reasonably quick to respond to the user's http request to the server.
- The webpage that is provided after the server request should present itself as a visually appealing and easily readable interface.
- The application should be written to account for both webpage and phone users in terms of UX.
 - If the webpage is opened up on a phone, the layout will be smaller and simpler, only displaying the essential information such as the capacity, and will have fonts [anaesthetics] set to basics so most phones no matter the browser will be able to open it correctly.
- Safety will be based on the capacity of that section in the gym.

1.4 Interfaces

1.4.1 User Interface:

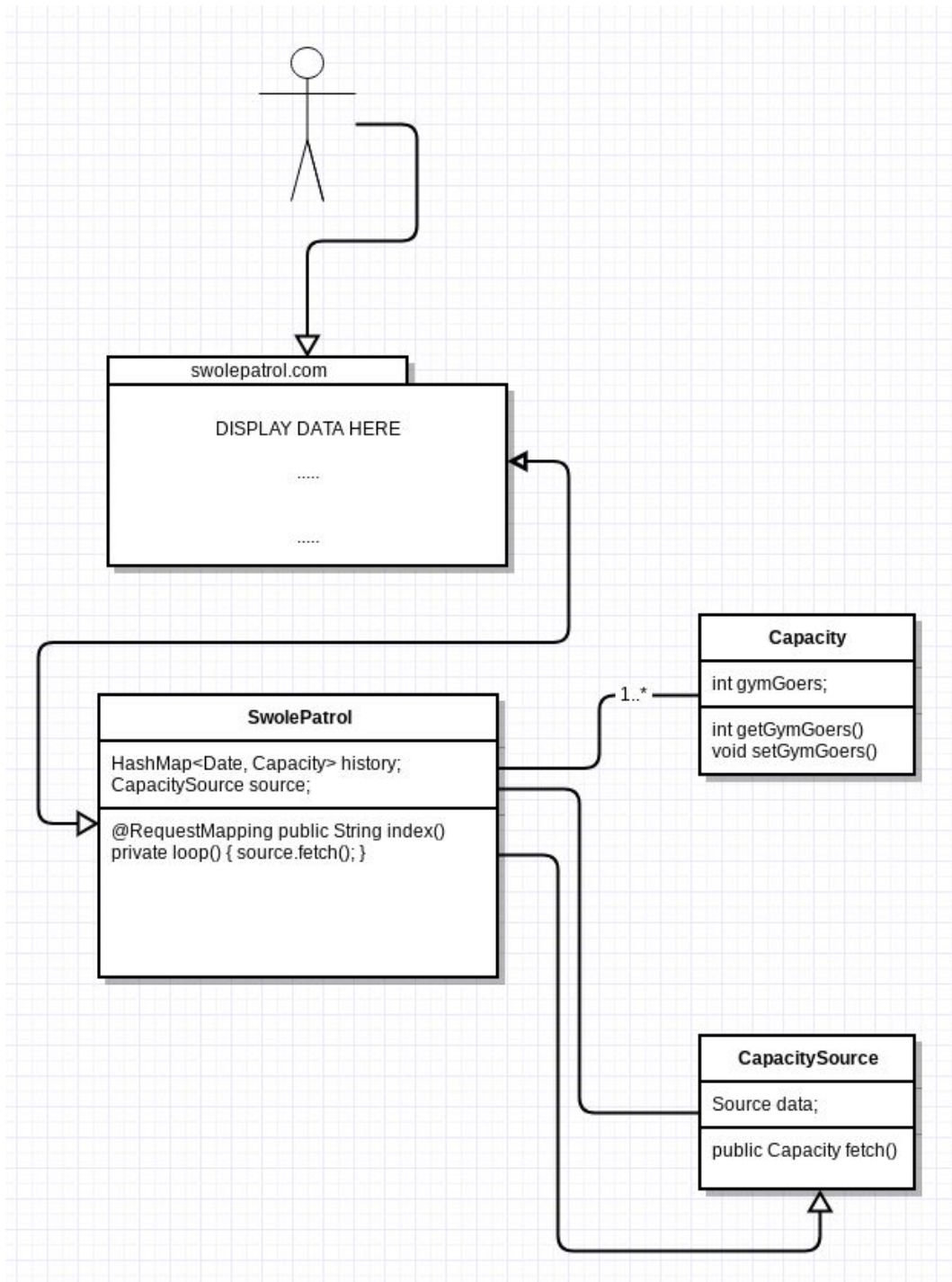
- When a user accesses the webpage they will see an estimated gym capacity as a percentage followed by the capacity of the other gym facilities (i.e. weight room, pool, basketball courts, etc...).
- If the user wants to see the estimated population they can click on the facility they would like to see which would show them a breakdown of the hourly population

1.4.2 Software Interface

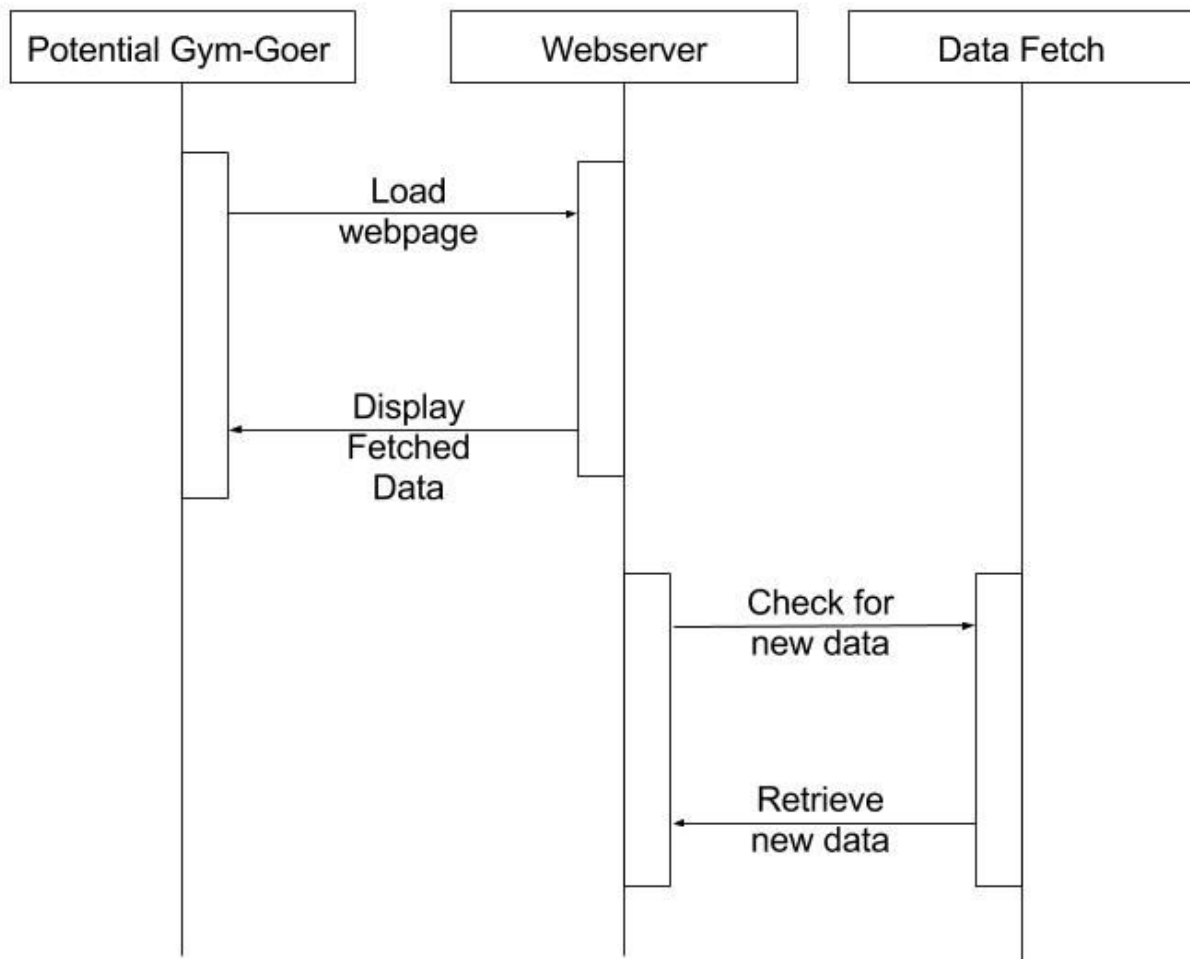
- The software will take in the excel documents and extract the data needed
- Once the data is input, the functions will generate the required statistics and estimated population

2 UML Design

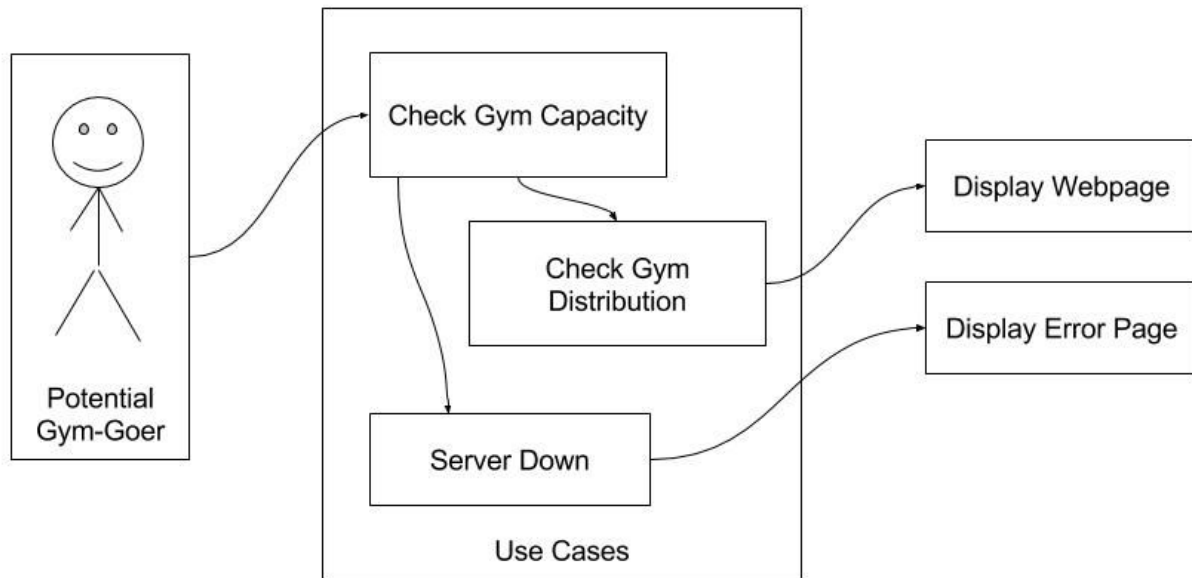
2.1 Classes



2.2 Sequence



2.3 Use Cases

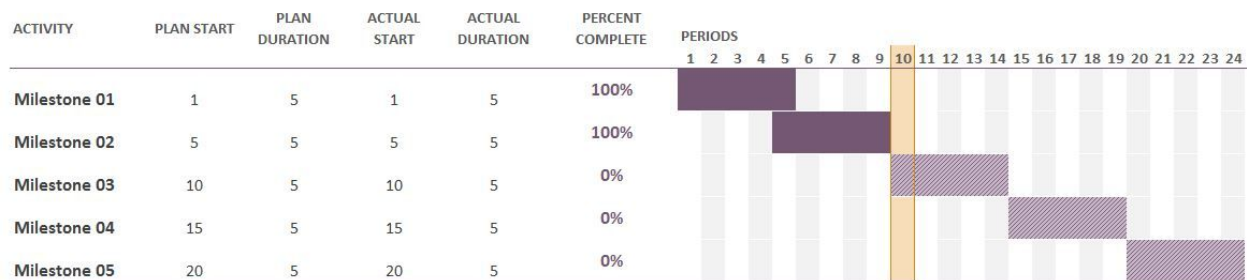


3 Test Plan

To test the validity of our results and implementation of our modeling functions, we will have to manually do the calculations and see whether the results match up. There are many statistical methods that could calculate the values we need, so we will implement them and calculate them ourselves to see if our implementation is accurate with our work. Although this testing will be tedious, our entire project will depend on the validity of our values so it is worth the extra time to verify our results

4 Project Plan (Update)

- Milestone: Request from UNT gym to see how many people are checking. If they say no, we will need to collect data ourselves and derive our own model. Complete! (They said yes)
- Milestone: ~~Establish a working webpage in Java~~ Complete!
- Milestone: Implement the data/model into the webpage
- Milestone: Implement the functions that transform the data into presentable information for the user
- Milestone: Finalize the webpage and debug/ensure everything runs smoothly



5 Risk Management (Update)

5.1 Updated Risks

- UNT does not send us automated reports of data (they have agreed to give data)
- UNT's reports don't come in an easy format to implement
- The data does not work with our Java implementation (incompatible data types)
- Unable to build a solid model of gym population distribution (lack of knowledge)

5.2 Potential solutions

- If we can't get automated data feeds, then we'd have to manually insert the data ourselves
- Lookup statistical models and learn the equations needed to predict future population values
- If the reports are not in a nice easy format, we could manually change them into excel spreadsheets then implement them.

6 Meeting Minutes

	A	B	C	D
1	Date	Time	Participants	Group Progress Description
2	9/12/2017	3PM-3:40PM	Hanson Nguyen El-Rayes Hicks	- Defined our 5 risks. - Worked on our Deliverable 1 report as a group, broken into sections. - Shed light on the internal problems we as a group might have.
3	9/19/2017	3:20-3:50PM	Hanson Nguyen El-Rayes Hicks	- Established a connection with the assistant director to obtain the data over Email. - Converse with each other on how to make the data more precise and accurate.
4	9/26/2017 & 9/28/2017	3 - 3:50PM & 2:55 - 3:50PM	Hanson Nguyen El-Rayes Hicks	- worked on creating the backbone of the program; by using java eclipse we can upload data to a website. - Taked about how the distribution model should look under given circumstances.
5	10/3/2017	3:10- 3:50PM	Hanson Nguyen El-Rayes Hicks	-Brainstormed and created an outline for certain sections in deliverable 2. -Finished the Use case chart and Sequence chart for deliverable 2. -Talked about user design and how data will be presented.
6	10/5/2017	2:45 - 3:50PM	Hanson Nguyen	- Got familiar with BootStrap in order to start working on the user interface using HTML CSS and JS framework.
7	10/10/2017	3:10 - 3:50PM	Hanson Nguyen El-Rayes Hicks	-Splitting each section among ourselves, we finished Dilverable 2.

7 Project Status

Our project is right where it should be. The conversation between us and the staff at the Pohl Recreation Center is going well, and we are making progress getting *live* data for the application.

After we finish collecting the necessary API components, we will begin implementing the necessary framework to allow for a web server to provide our desired dataset. Our plan to build the application to our desired specifications is still on track, and none of our risks have presented themselves as being problematic.

8 Member Contribution

Anthony 25%:

- Updated project plan
- Contributed to requirements

Ibrahim 25%:

- Updated Risks and their potential solutions
- Wrote the interface chapter

John 25%:

- Created functional requirement and Contributed some to nonfunctional requirements.
- Recorded and updated the time meeting sheet.

Jacob 25%:

- Created original deliverable document
- Setup document template / format
- Managed repository to contain the necessary files and structure