NORTHERN ARIZONA UNIVERSITY

Mobile Data Collection for Gait Analysis

Team MDC

April 29, 2016

David Nagel, Jack Burrell,
Ahmad Meer, Justin Poehnelt

Project Mentor: Dr. Omar Badreddin

Project Sponsor

Dr. Kyle Winfree

- Department of Informatics and Computing
- The PD Shoe is designed to make simple reminders for patients with Parkinson's Disease.

PhD, Biomechanics and Movement Science

WEARABLE INFORMATICS LABORATORY

Data Collection for Gait Analysis

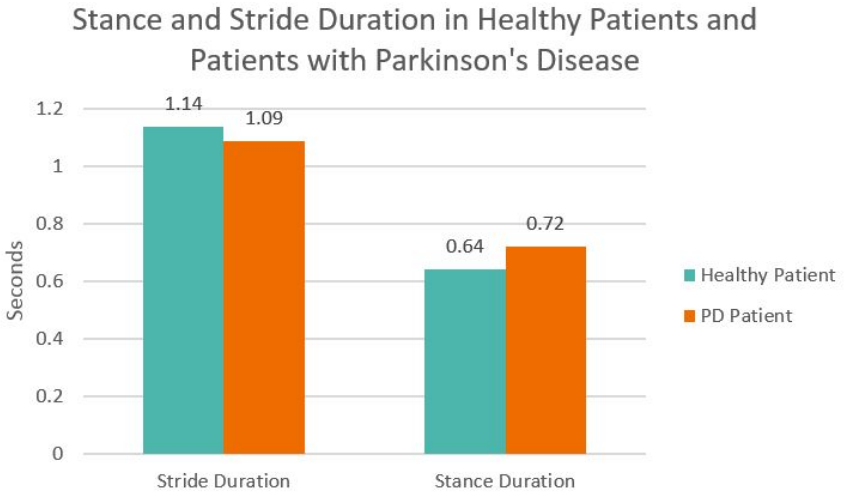
Few commercial tools for collecting data on gait. Data often limited to activity level.

Existing Wearable Devices:

- Fitbit
- Jawbone
- Nike+ Sportband
- LifeGait

Data Collection for Gait Analysis

- Raw data collection outside of clinical setting allows for analysis outputs
  - Stride Duration
  - Foot Strike Pattern
  - Weight Distribution
- 10 million patients worldwide with PD
- Supports diagnosis and testing of treatment effectiveness and other physical therapies



Challenging Requirements

Requirements elicited from regular meetings with sponsor:

- Sufficient Granularity of Data
  - Time Delta Between Readings of less than 10 milliseconds
- Near Real-time Analysis of Data
- Automated Data Centralization
- Access to Data for Statistical Packages such as Matlab and Octave
- Standardized Modules Extensible to Many Wearable Devices

Key Risks

Loss of data
Power interruptions, network congestion, weak control flow system

Poor Data Granularity
Inefficient hardware or software design

Lag in Data Availability
Poor network connectivity

Data Synchronization Errors
Unsynchronized devices and poorly calibrated sensors

Server Cannot Handle Number of Requests
Potentially 1000 rows to insert every 10 seconds

Postgresql limits
Max Table Size - 32 TB

Development Process

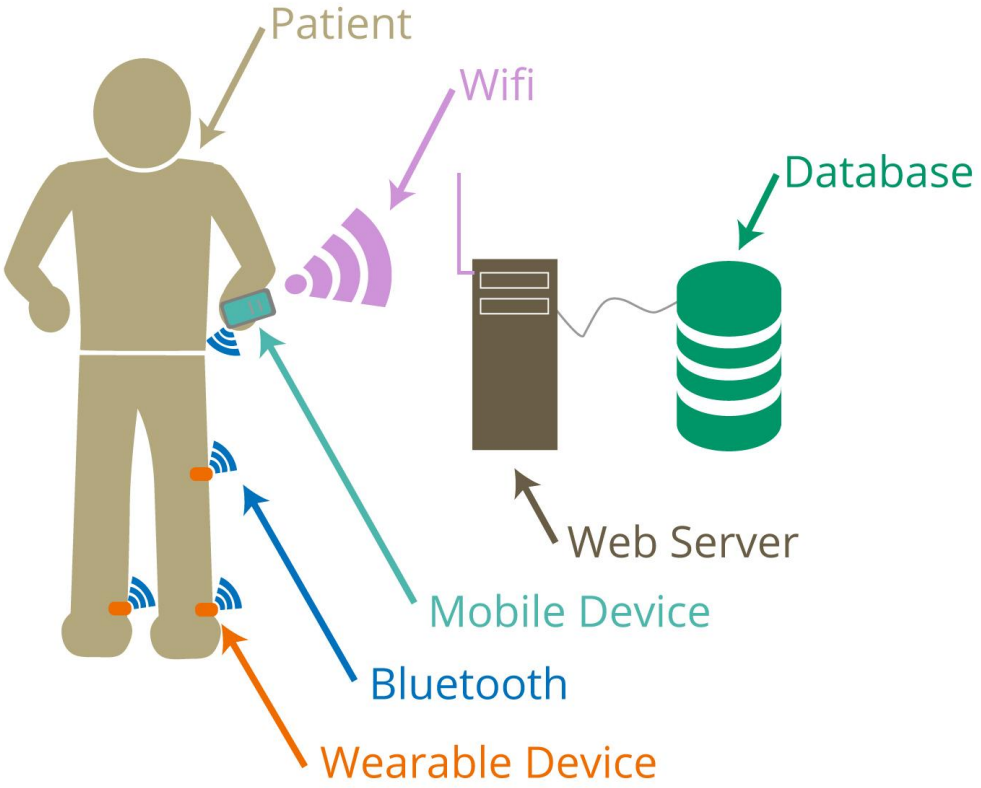
Evolutionary Rapid Development

WBS	Tasks	Start	End	Duration (Days)	% Complete	Working Days	Days Complete	Days Remaining
1	**Milestone 1 (Hardware Level)**	1/16/16	3/17/16	62	100%	44	62	0
1.1	Assemble Hardware	1/16/16	1/21/16	6	100%	4	6	0
1.2	Develop Sensor Reading Capability	1/22/16	2/5/16	15	100%	11	15	0
1.3	Develop Basic Command Handling	1/22/16	2/16/16	26	100%	18	26	0
1.4	Develop Hardware Data Strucutre/SD	1/22/16	2/5/16	15	100%	11	15	0
1.5	Develop Advanced Command Handling	3/1/16	3/17/16	17	100%	13	17	0
2	**Milestone 2 (Mobile Application Level)**	2/17/16	4/6/16	50	100%	36	50	0
2.1	Design Application Protocol for Bluetooth	2/17/16	2/24/16	8	100%	6	8	0
2.2	Develop Android Application Frame	2/17/16	2/20/16	4	100%	3	4	0
2.3	Develop Testing User Interface	2/17/16	2/22/16	6	100%	4	6	0
2.4	Develop Android Bluetooth Compatability	2/21/16	3/3/16	12	100%	9	12	0
2.5	Implement Application Protocol into Android Application	3/4/16	3/15/16	12	100%	8	12	0
2.6	Develop Phone Database	3/16/16	3/27/16	12	100%	8	12	0
2.7	Develop Final User Interface	3/16/16	4/6/16	22	100%	16	22	0
3	**Milestone 3 (Web Server Level)**	1/30/16	4/6/16	68	100%	48	68	0
3.1	Setup Server Database	1/30/16	2/7/16	8	100%	5	8	0
3.2	Develop Server Database	3/11/16	3/17/16	7	100%	5	7	0
3.3	Develop Server API	3/18/16	4/6/16	20	100%	14	20	0
3.4	Develop HTTP Format	3/18/16	3/31/16	14	100%	10	14	0
4	**Milestone 4 (Testing & Refining)**	4/7/16	5/5/16	28	25%	21	7	21
4.1	Testing & Refining	4/7/16	5/5/16	28	25%	21	7	21

System Overview

Four Components

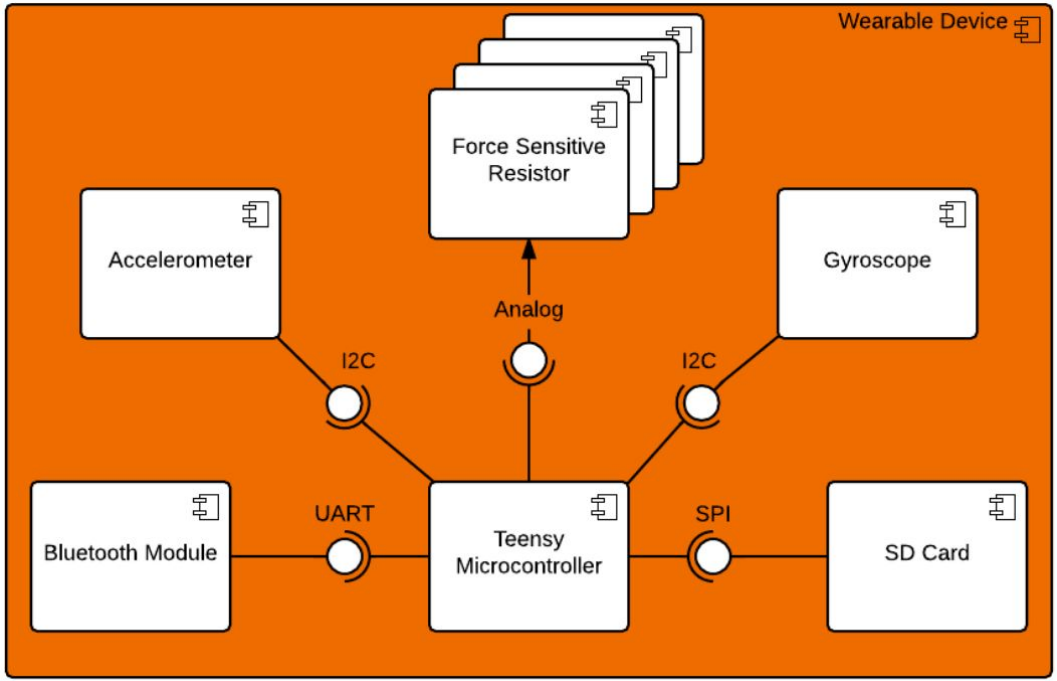
- Wearable Device
- Mobile Device
- Web Server
- Database



Wearable Device

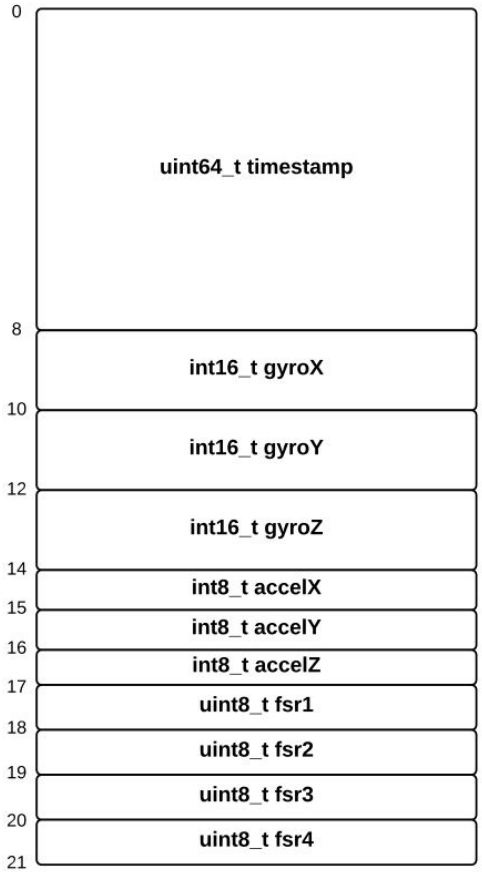
Teensy 3.2 microcontroller was selected over the Arduino and Photon

Modular design with various sensors and components



Wearable Device: Database

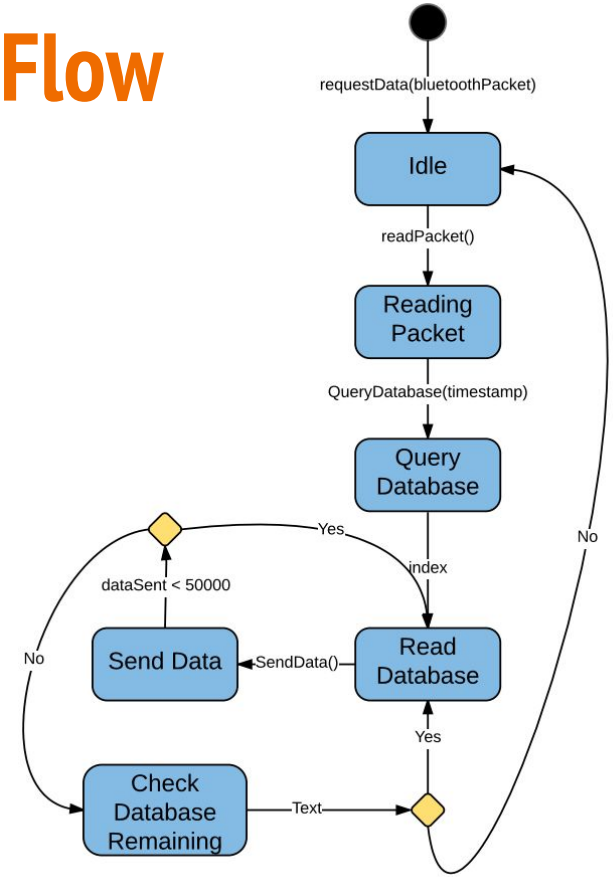
- Rows ‘indexed’ by timestamp
- Decoupled from specific sensors on device
- Space requirements reduced by 65% compared to CSV format



Wearable Device: Control Flow

Multiple Tasks

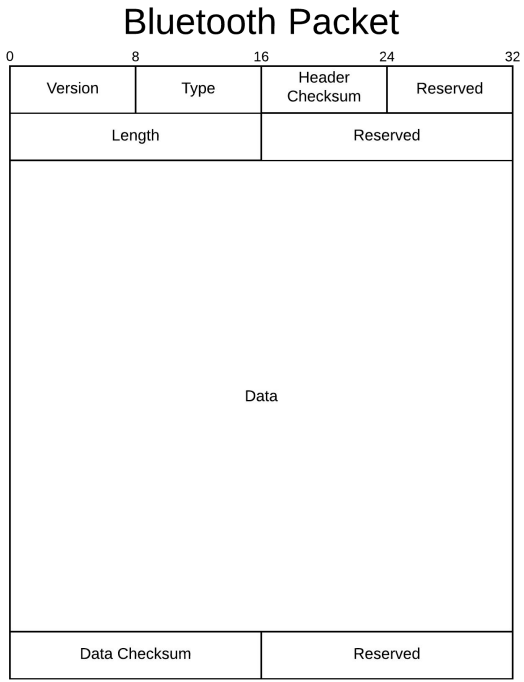
- Reading from sensors
- Send data to mobile device
- Time synchronization
- Status check
- Stop and start



Wearable Device: Communication

Bluetooth Application Protocol

- Time synchronization across numerous devices
- Data requests and responses
- Device identification with LED and/or vibration
- Sensor activation and deactivation



Mobile Device

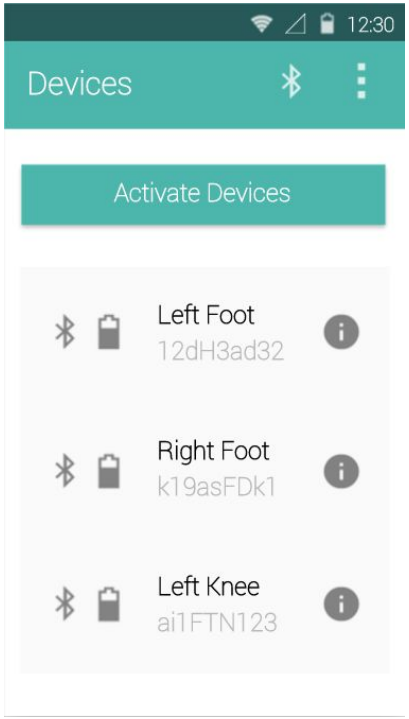
Bluetooth communication with the wearable device

WiFi connection to web server

Requests data from wearable device since last retrieved timestamp

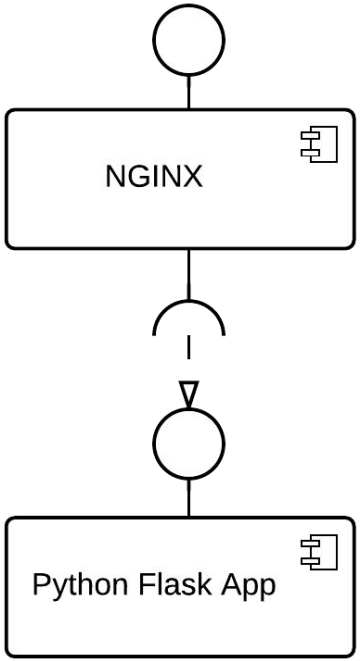
Caches data in a sqlite database until WiFi connection is available

Limited to Android platform



Web Server

- NGINX reverse proxy
- Python Flask application framework
- Handles HTTP POST requests from mobile device

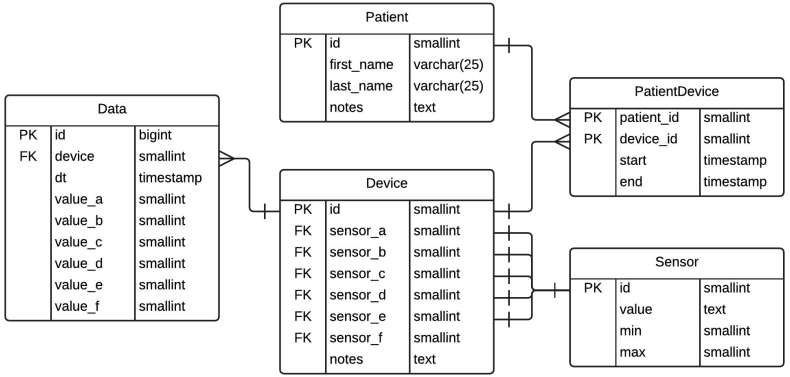


Database

Postgresql 9.3 has been selected for this system.

- Free and Open Source
- GIS Extensions
- Window functions for smoothing and cycle detection

Analysis may be completed using another layer above Postgresql depending on research needs.



Entity Relationship Diagram

Project Testing

Unit Testing Components & Integration Testing

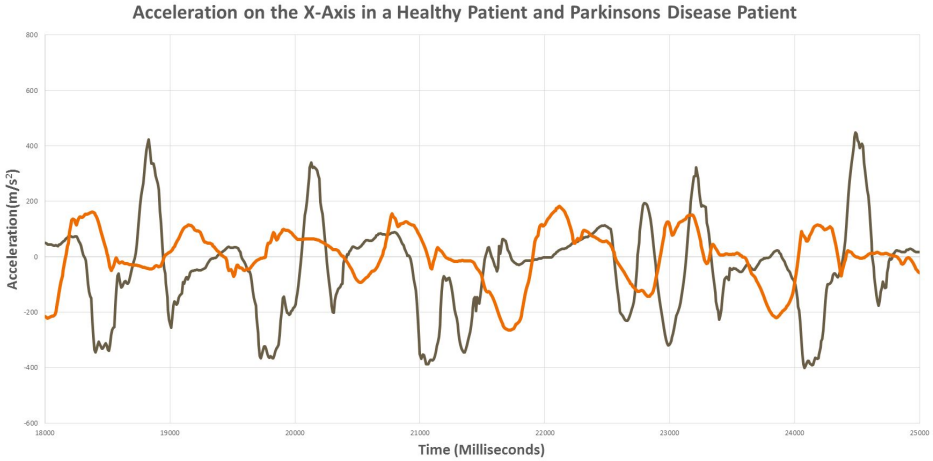
- Wearable device difficult to test due to limited emulation options
- Other components have available libraries and interface mocking

Functional Testing

- Data successfully transferred through hierarchy of system
- Performance and efficiency tests of data communication

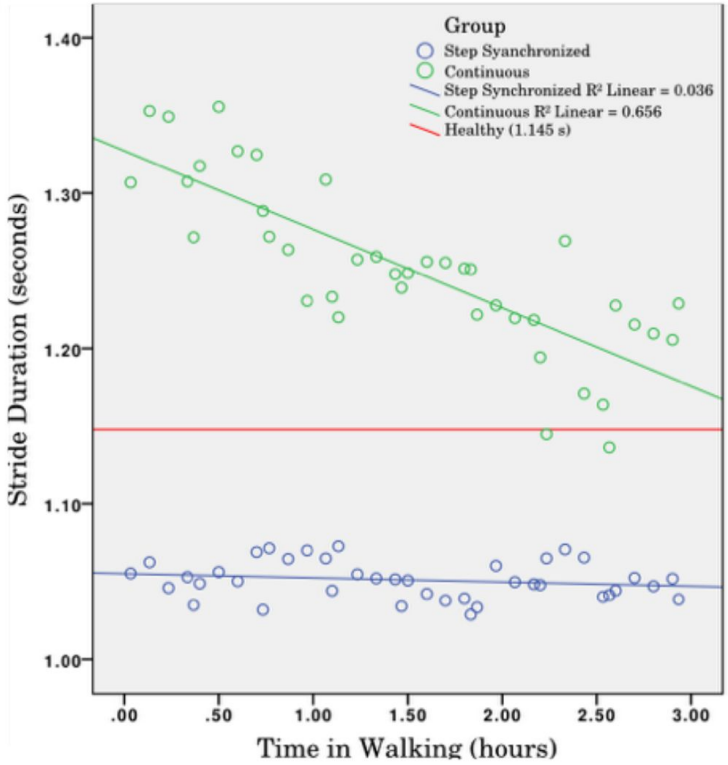
Non-Functional Testing

- Usability testing with Dr. Winfree's research students
- Evaluate hardware reliability in shoe form



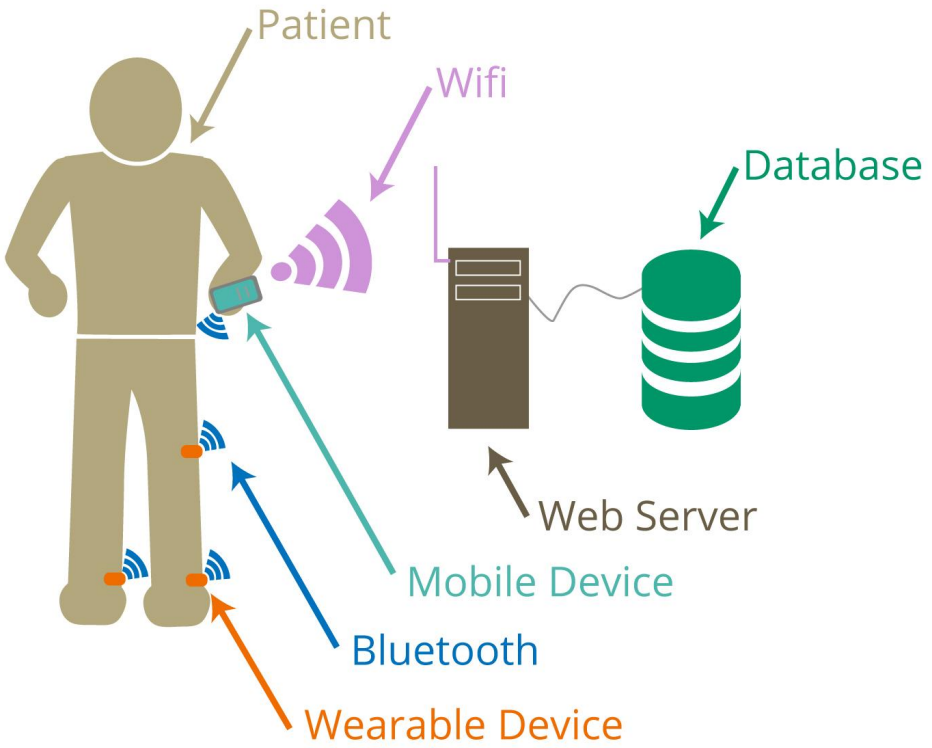
Future Work

- Statistical Analysis for Detecting Current Activity
- Optimization
  - Power efficiency
  - Bluetooth efficiency
- Data Analysis and Visualization Through Web API
- Embed Wearable Device in Shoe



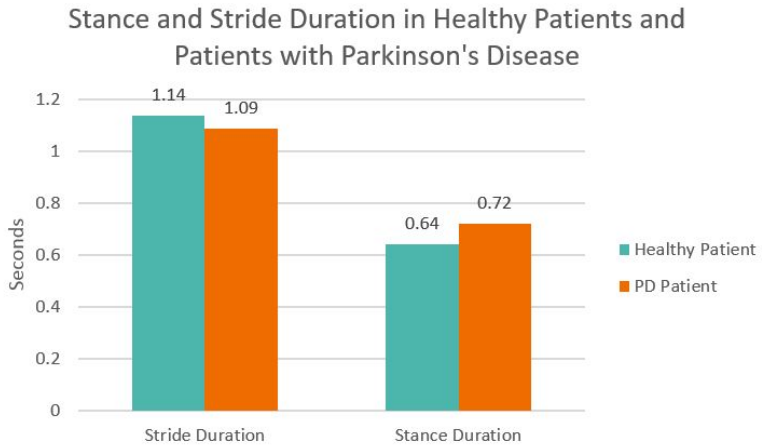
Conclusion

Centralized and near real-time collection of data for gait analysis to assess treatment impact and improve early diagnosis of Parkinson's Disease

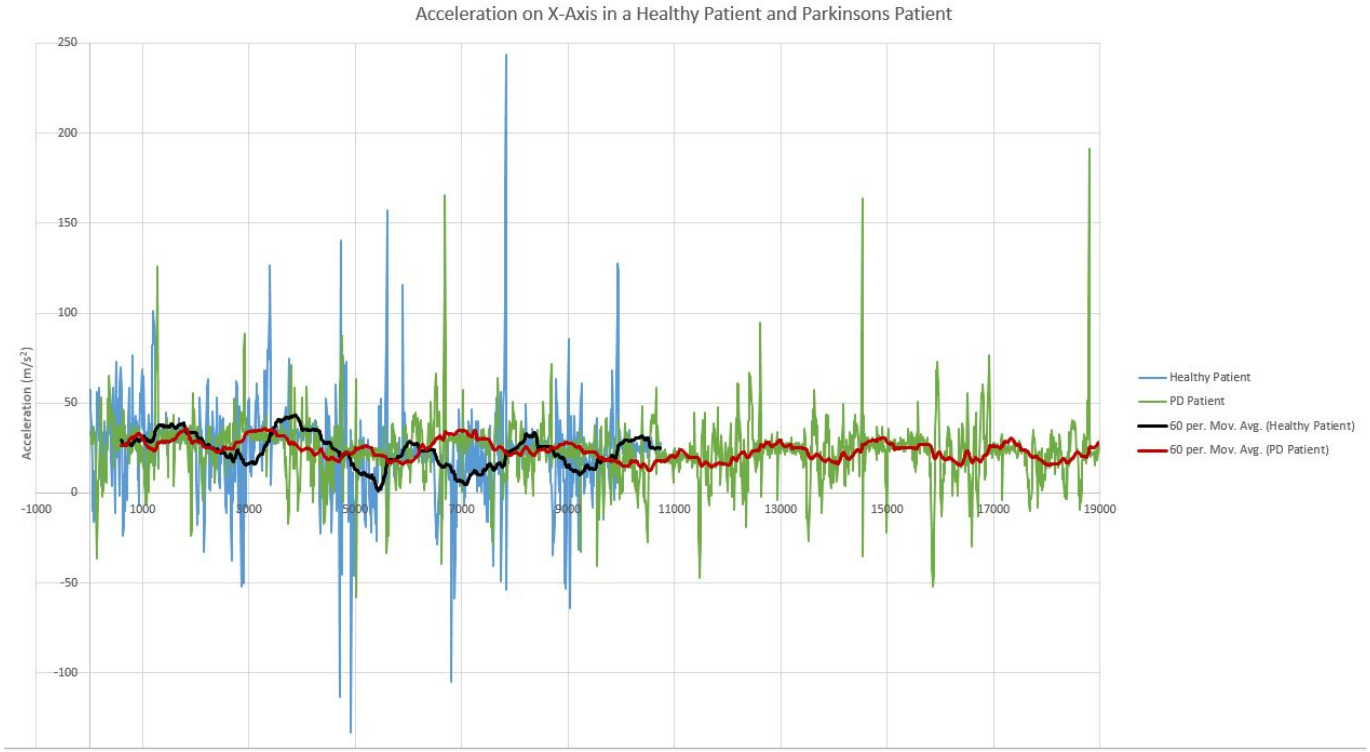


Analysis Outputs

	Healthy	PD
μ Stride Duration (s)	1.14 (0.11)	1.09 (0.19)
σ Stride Duration (s)	0.05 (0.02)	0.07 (0.03)
μ Stance Duration (s)	0.64 (0.12)	0.72 (0.14)
σ Stance Duration (s)	0.08 (0.06)	0.10 (0.06)
μ Stance Duration (%GC)	56.00 (7.51)	66.22 (6.11)
σ Stance Duration (%GC)	5.54 (5.00)	8.16 (6.64)
μ Heel Max (%GC)	10.19 (4.73)	16.72 (7.00)
σ Heel Max (%GC)	3.65 (1.48)	8.37 (6.61)
μ Ball Max (%GC)	42.15 (8.60)	44.17 (6.38)
σ Ball Max (%GC)	6.45 (5.81)	7.38 (6.97)
μ Toe Max (%GC)	53.12 (2.30)	52.44 (6.44)
σ Toe Max (%GC)	3.06 (1.99)	6.60 (6.79)



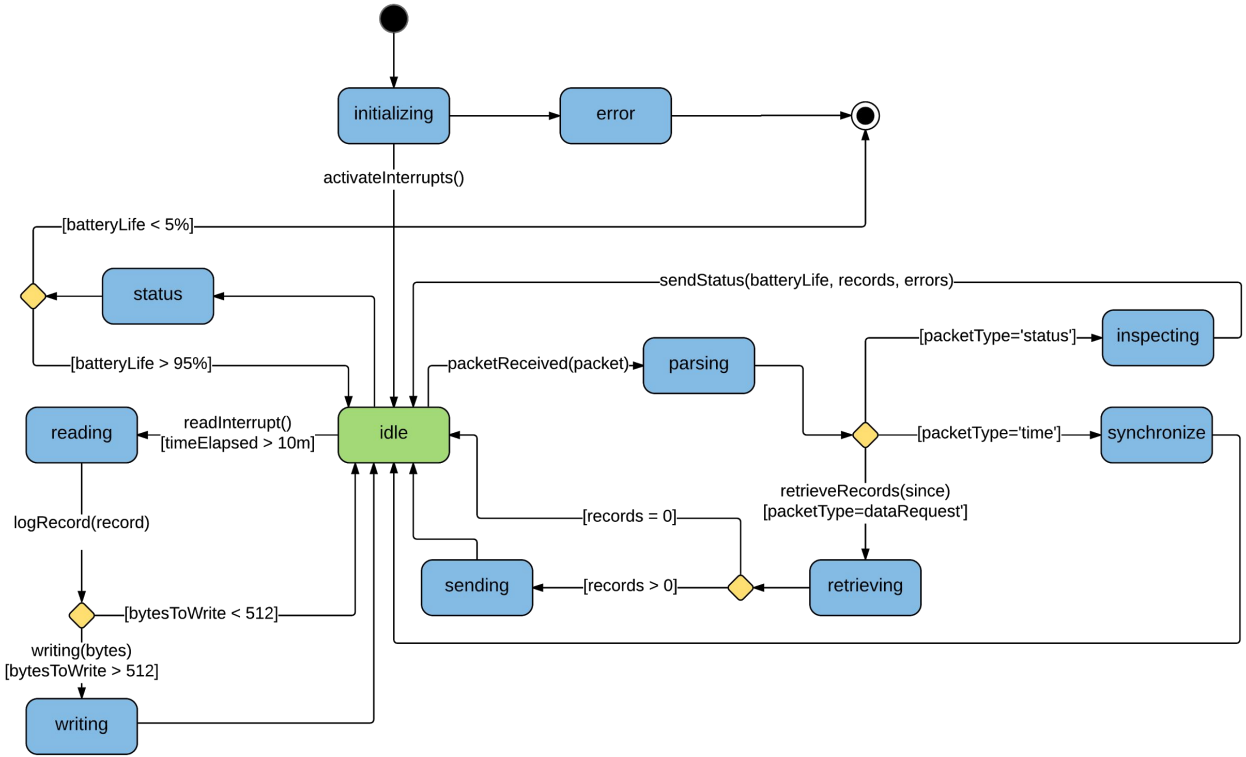
Sample Data

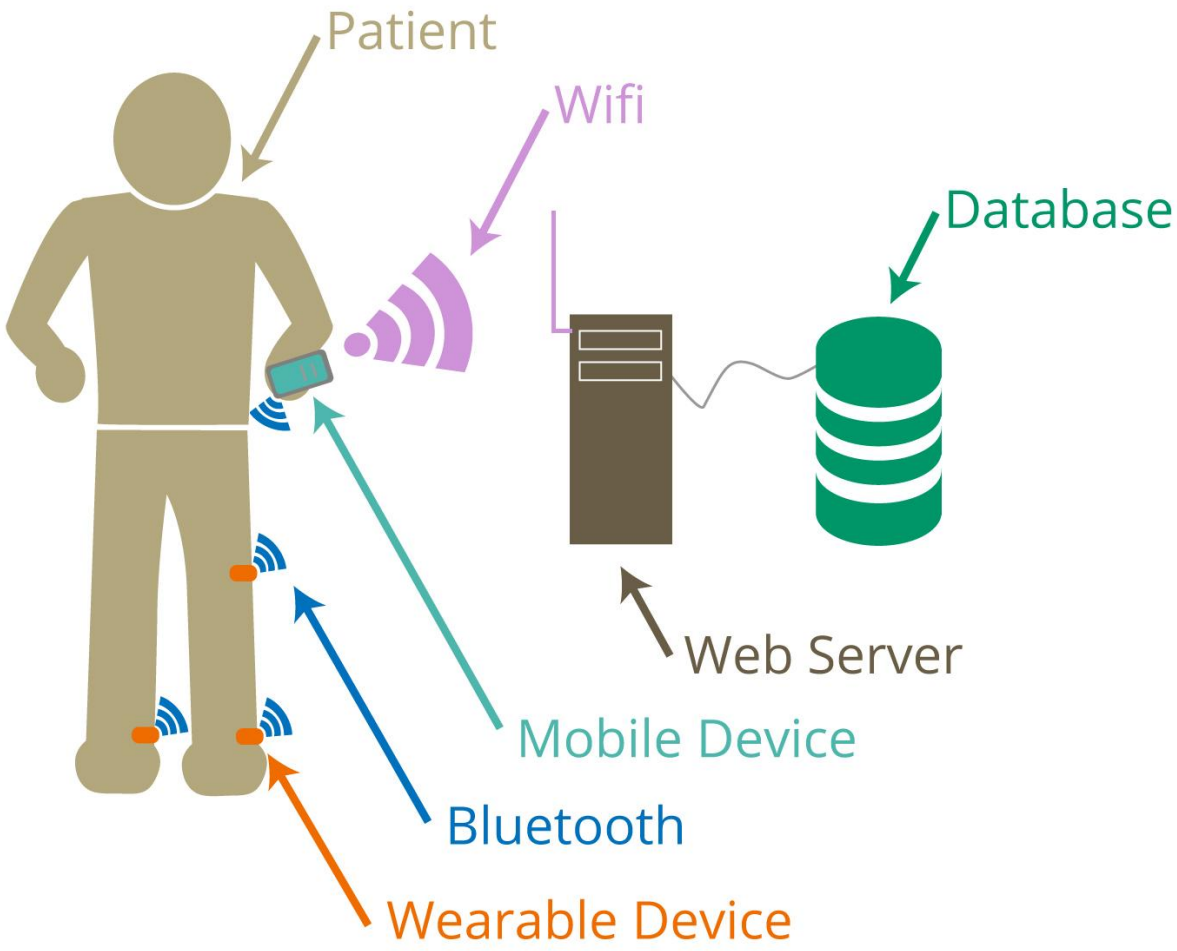

Acceleration on X-Axis in a Healthy Patient and Parkinsons Patient
Acceleration (m/s²)
250
200
150
100
50
0
-50
-100
-1000
1000
3000
5000
7000
9000
11000
13000
15000
17000
19000
Healthy Patient
PD Patient
60 per. Mov. Avg. (Healthy Patient)
60 per. Mov. Avg. (PD Patient)

Example Data

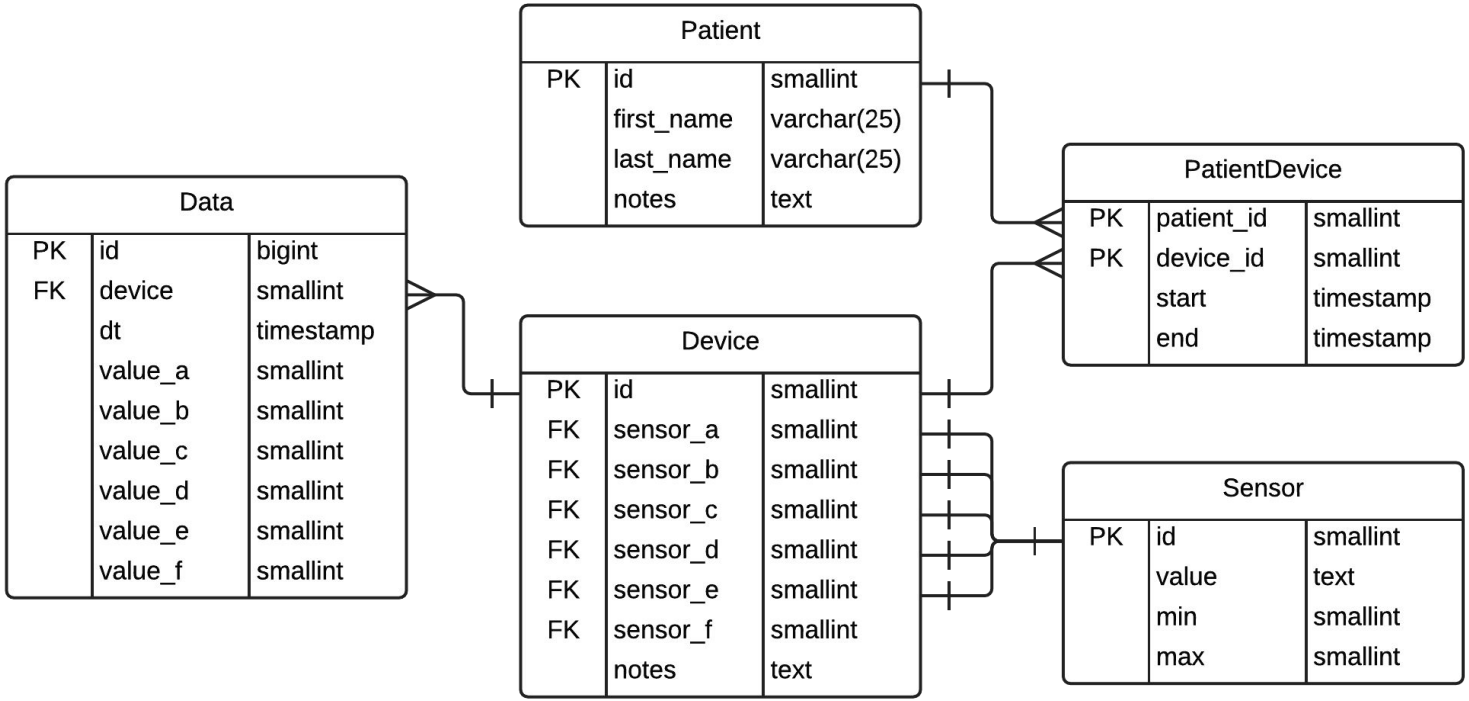
Time	FSR1	FSR2	FSR3	FSR4	AccelX	AccelY	AccelZ	GyroX	GyroY	GyroZ
8502	13	1	1	13	0.00	0.11	1.16	-0.51	-0.69	-0.71
9107	14	1	1	13	-0.00	0.11	1.17	-0.51	-0.69	-0.71
9711	9	2	2	17	0.01	0.11	1.14	-0.51	-0.69	-0.71
10315	14	1	1	2	0.00	0.12	1.16	-0.51	-0.69	-0.71
10919	1	1	1	3	0.00	0.11	1.15	-0.51	-0.69	-0.71
11523	5	1	1	11	0.01	0.12	1.15	-0.51	-0.69	-0.71
12128	16	1	1	2	0.00	0.11	1.16	-0.51	-0.69	-0.71
12732	2	1	1	2	0.00	0.11	1.15	-0.51	-0.69	-0.71

Wearable Device: Control Flow




Patient
Wifi
Database
Web Server
Mobile Device
Bluetooth
Wearable Device

Entity Relationship Diagram

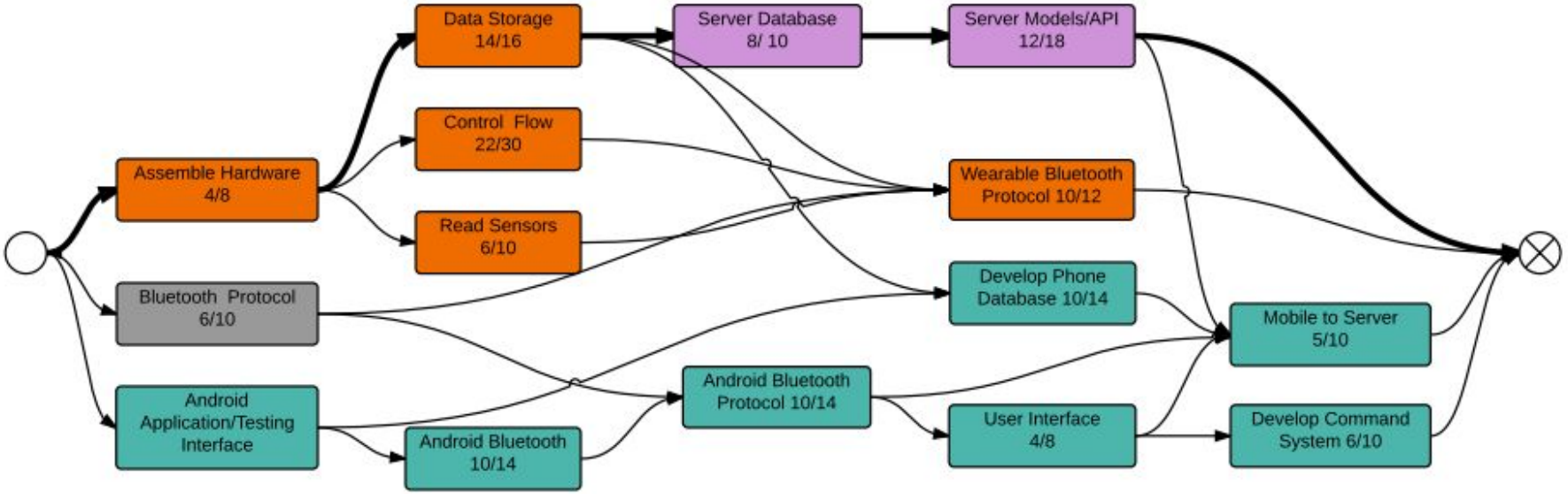


Schedule

PERT Chart used to determine three major milestones

1. Wearable device assembled, collecting data, and communicating (In testing)
2. Mobile application receiving and storing data (In testing)
    a. Performance issues over bluetooth
    b. GUI finalized
    c. Automation
3. Web server receiving and storing data (In testing)
4. Unit Testing

Pert Chart


Data Storage
14/16
Server Database
8/ 10
Server Models/API
12/18
Control Flow
22/30
Assemble Hardware
4/8
Wearable Bluetooth
Protocol 10/12
Read Sensors
6/10
Develop Phone
Database 10/14
Bluetooth Protocol
6/10
Mobile to Server
5/10
Android
Application/Testing
Interface
Android Bluetooth
Protocol 10/14
Android Bluetooth
10/14
User Interface
4/8
Develop Command
System 6/10

Schedule and Effort Estimation

WBS	Tasks	Start	End	Duration (Days)	% Complete	Working Days	Days Complete	Days Remaining
1	**Milestone 1 (Hardware Level)**	1/16/16	3/17/16	62	100%	44	62	0
1.1	Assemble Hardware	1/16/16	1/21/16	6	100%	4	6	0
1.2	Develop Sensor Reading Capability	1/22/16	2/5/16	15	100%	11	15	0
1.3	Develop Basic Command Handling	1/22/16	2/16/16	26	100%	18	26	0
1.4	Develop Hardware Data Strucutre/SD	1/22/16	2/5/16	15	100%	11	15	0
1.5	Develop Advanced Command Handling	3/1/16	3/17/16	17	100%	13	17	0
2	**Milestone 2 (Mobile Application Level)**	2/17/16	4/6/16	50	100%	36	50	0
2.1	Design Application Protocol for Bluetooth	2/17/16	2/24/16	8	100%	6	8	0
2.2	Develop Android Application Frame	2/17/16	2/20/16	4	100%	3	4	0
2.3	Develop Testing User Interface	2/17/16	2/22/16	6	100%	4	6	0
2.4	Develop Android Bluetooth Compatability	2/21/16	3/3/16	12	100%	9	12	0
2.5	Implement Application Protocol into Android Application	3/4/16	3/15/16	12	100%	8	12	0
2.6	Develop Phone Database	3/16/16	3/27/16	12	100%	8	12	0
2.7	Develop Final User Interface	3/16/16	4/6/16	22	100%	16	22	0
3	**Milestone 3 (Web Server Level)**	1/30/16	4/6/16	68	100%	48	68	0
3.1	Setup Server Database	1/30/16	2/7/16	8	100%	5	8	0
3.2	Develop Server Database	3/11/16	3/17/16	7	100%	5	7	0
3.3	Develop Server API	3/18/16	4/6/16	20	100%	14	20	0
3.4	Develop HTTP Format	3/18/16	3/31/16	14	100%	10	14	0
4	**Milestone 4 (Testing & Refining)**	4/7/16	5/5/16	28	25%	21	7	21
4.1	Testing & Refining	4/7/16	5/5/16	28	25%	21	7	21

Slack Time

Critical Path

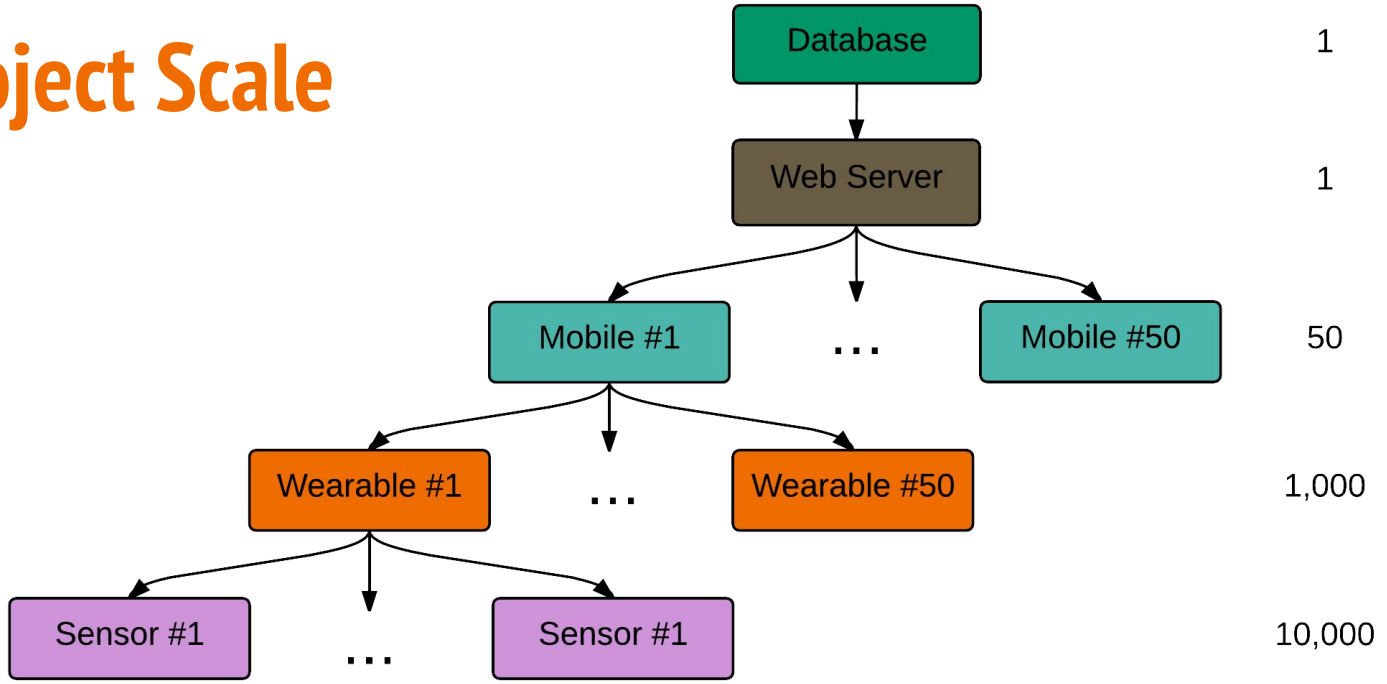
1. Assemble Hardware
2. Data Storage
3. Server Database
4. Server API
5. Web Application

Estimated Days: 58

Assumes days for task is average of best and worst case estimates. Some tasks may require multiple team members.

		Start Day		
	Days	**Earliest**	**Latest**	**Slack**
Wearable Device				
Assemble Hardware	6	0	0	0
Data Storage	15	6	6	0
Control Flow	26	6	21	15
Read Sensors	8	6	39	33
Bluetooth Protocol	11	32	47	15
Design				
Bluetooth Protocol	7	0	23	23
Mobile Device				
Android Application	4	0	14	14
Android Bluetooth	12	4	18	14
Android Bluetooth Protocol	12	16	30	14
Mobile Database	12	21	38	17
Mobile to Server	8	45	50	5
User Interface	6	28	42	14
Data Analysis on Mobile	10	34	48	14
Command System on Mobile	8	34	50	16
Server and Database				
Server Database	9	21	21	0
Analysis Queries	8	30	37	7
Server Models/API	15	30	30	0
Web Application	13	45	45	0

Project Scale

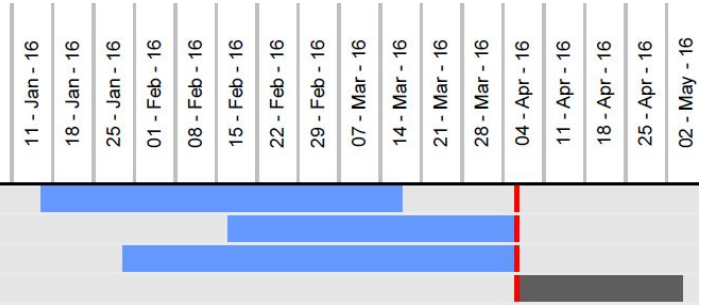


100 readings a second for 30 days on 1,000 Wearable Devices = 172 Billion Rows/month

30 bytes captured 100 times a second on 1,0000 Wearable Devices = 3 MB/second

Schedule Estimation and Effort Estimation

WBS	Tasks	Start	End	Duration (Days)	% Complete	Working Days	Days Complete	Days Remaining
1	Milestone 1 (Hardware Level)	1/16/16	3/17/16	62	100%	44	62	0
2	Milestone 2 (Mobile Application Level)	2/17/16	4/6/16	50	100%	36	50	0
3	Milestone 3 (Web Server Level)	1/30/16	4/6/16	68	100%	48	68	0
4	Milestone 4 (Testing)	4/7/16	5/5/16	28	0%	21	0	28



Project Status

Core functional requirements from project sponsor have been achieved.

Remaining Tasks

- Performance Issues with Bluetooth Communication
- Mobile Device GUI Revision
- Automation of Mobile Device Functionality
- Continued Testing
- Data Analysis and Validation

NORTHERN ARIZONA UNIVERSITY

Theme Colors

ef6c00

4db6ac

ce93d8

695d46

b3a77d

009668

3. Show fitbit and other devices are a failutre

- Get data and show it
- Actual Images
- How this is could be a big step in the medical world

4. System Design

- Make sure to show what kinds of communication
- Use UML (or any other standard)
- Bigger Font
- Make it clearer with pictures
- fill the slide
- SHow the patient

8. Be ready to defend why we have the intermediate device

9.

- ER Diagram
- Why is this suitable for a write heavy system?
- Move 10 TB to the tree slide
- What kind of licensing

10. Be sure to know what D3.js is
    - Understand why JSON is inefficent and be prepared to defend it (or stay open to future changes)
11. How many days before the SD card will fill
12. Increase Font size
13. Rename “Key Risks”
15. Back up slides
16. 4 key points

Overall

- Communicate the software side of this
- Is this Agile or Rapid Prototyping?
- Why we choose certain pieces of hardware over others through the whole presentation.

Notes from last time

NORTHERN ARIZONA UNIVERSITY

- Saying big data is a distraction
- Start with small data set
- Store all data?
- Schedule was confusing
- Keep in mind the number of patients within out scope
- Make the beginning very clear on the clinical setting aspect
- Some of the risks out of our range
- Possibly create our own database for the size?
- The graphic for the database does not show the right database (If I remember correctly this comment had something to do with using postgressql and an ERD diagram)
- Milestones are requirements. We can use them again in the presentation, we introduce them late.
- Intro was a bit long
- solution overview was a bit long
- more detail on requirements acquisition (who knows how much for design review 2)

Pretty the comments from last time, from what I gathered were more about style of presenting. Other than the focus on big data of course which makes everyone angry.

I inserted pictures of the presentation description after this so we don't have to go looking through Otte's website everytime

This time around we have more specific diagrams for our architecture and such

CS486c - Design Review II

Overview

This assignment will have you prepare and deliver your second design presentation to the CS faculty and CS486C students.

Introduction

A design review is a mechanism for presenting your progress thus far and receiving feedback from your fellow capstone students and the computer science faculty. This feedback will cover all aspects of your work, from your project concept, to your understanding of the requirements, to design decisions you've made, to the architecture, to your progress toward implementation. In addition, feedback will address the effectiveness of your presentation materials and delivery.

Presentation Structure

Your presentation should have the following structure:

Introduction

Introduce your team (and team name) and your team members, as well as any specific responsibilities they may have (lead tester or lead customer contact, for example).

Problem Statement

You should once more present a short overview of the problem you are addressing through your development work on your project: Who is your sponsor? What problem domain do they work in? What is the specific problem or challenge that they need solved (the core goal of your project)?

Your discussion here should focus on the specific challenges that your customer faces: So, you shouldn't say "Well, our customer wants a FooBar generator, so we're building one." Instead, you should say "FooBar generation is important because it supports the search for alien life; generating foobars is also challenging because foos are elusive and don't get along with bars." You get the idea: Make the audience believe that the problem you are addressing is both important and challenging.

Graphics are invaluable in this section: Think of images that exemplify the problem domain or diagrams that make the problem clearer, and use these graphics to help deliver your point during your presentation.

Solution Overview

With the background for your project established, you should then (once more) present a short overview of your solution. This should be a relatively high-level discussion focused on all systems (not just software) involved in your project: What is the system being developed? How will the customer interact with or use this system? What other software systems does your project interact with? What are the sources of information that are used?

Your solution overview should suggest that you have a deep understanding of the problem, you know the advantages and disadvantages of other possible ways that it could be solved, and have a compelling overall vision. This part of your presentation should invoke confidence in the audience about your proposed solution and spark interest in the rest of your talk.

Key Requirements

For this section, you should once more briefly discuss the key functional and non-functional requirements for your project. Keep in mind that you should be discussing the most interesting and unique of these requirements -- the ones that differentiate your system from others. For example, requirements on your login/logout system and the associated need to maintain user account information are not very interesting, as this is a common feature-set---mention something like this briefly, and move on. Requirements on the unique functionality and performance requirements of your FooBar generation algorithm, however, are much more interesting. Also keep verifiability in mind when discussing non-functional requirements: saying "Our FooBar generation must work fast" is not very useful, but "The FooBar algorithm shall complete in 1.7 seconds for one thousand foos and bars" is better.

Development Methodology

With an understanding of your requirements in place, next spend a short amount of time talking about the software development methodology you're using: What lifecycle model have you adopted? What features of your system, such as its functional or non-functional requirements or problem domain, motivated your selection? How do your team members coordinate development activities? What tools have you chosen to support your methodology? Showing screenshots of commit logs or task- and feature-tracking systems that you're using would be great.

Architecture

Present an overview of your software architecture, showing the top-level components involved and how they're organized. Discuss the main responsibilities of each component, and cover the main information flows (and their format) that convey data from one component to another. Discuss any architectural style or design pattern influences that shaped your architecture, and why these styles or patterns were appropriate.

Implementation

Discuss your progress in implementing your system, which should be well underway. Present the most interesting aspects of the programming experience, such as particularly interesting algorithms, data structures, or integration with other software systems, and discuss the specific assignments that each team member had in coding. Most importantly, show your product thus far: Present screenshots or video capture of your project---you can do a live demo, if you so wish, but make sure it's 110% stable if you do.

Challenges and Risks

Provide an update on the important challenges and risks in your project: Which challenges have you already overcome? Which ones remain?

For each of these, make sure to discuss (a) the probability that the risk or challenge manifests, (b) the effect or danger should the risk or challenge surface, and (c) what you will do to mitigate this. Remember that these risks should be relevant to the project and your customer, not your own failures. In other words, discussing about how a risk is that "Our team will not be able to learn fast enough" or "Our team will not be able to finish in time" is entirely inappropriate. Think of how your software could affect the customer, the system's users, and those that rely on your user's actions.

Schedule

Provide an update of your progress by referring back to your project plan, and discussing which tasks are on- or off-schedule.

Conclusion

Finish your talk by providing a solid summary of your presentation: This should touch on the key points from your Problem Statement and Solution Overview, with a discussion of the main architectural and implementation take-away points. You want to end the presentation with an energetic note that conveys professionalism and justified confidence.

Guidelines

The entire presentation should take no longer than 15 minutes -- this is a sharp cut-off, and one of the faculty will cut you off when you reach that point. You've already presented information on a lot of the sections indicated above, so this presentation is weighted toward new material. The time you should invest in each section should be roughly as follows:

Section	Time
Introduction	No more than 30 seconds
Problem Statement	Roughly 2 minutes
Solution Overview	Roughly 1 minute
Key Requirements	Roughly 1 minute
Development Methodology	Roughly 1 minute
Architecture	Roughly 3-4 minutes
Implementation	Roughly 3-4 minutes
Challenges and Risks	Roughly 1 minutes
Schedule	Roughly 1 minute
Conclusion	Roughly 1 minute

Every member of your team should participate and contribute to your presentation. You may, of course, allocate who you think are your most capable public speakers on the most critically important parts of your talk.

Some tips for success:

- *Format*: Use large fonts, plenty of bullets, and short sentences. No more than 5-6 bullets per slide.
- *Polish*: Make sure your presentation is put together well. Spellcheck, check fonts, and align elements appropriately.
- *Prepare*: Practice your presentation and practice the hand-offs of the talk from team member to team member. When presenting, use the slides as your outline---if you have to read from your slides, you haven't prepared enough. Predict questions that you may be asked, and have answers ready.
- *Honesty*: The CS faculty are a pretty astute bunch and know when you're BS'ing. Be honest, particularly when you're asked questions, and don't be afraid to say "I don't know." Just make sure you know the answer next time.
- *Energy*: You should convey an impression that you're excited, energetic, and ready to work on this project.
- *Appearance and Conduct*: Dress and act professionally.

Schedule

You are expected to be in your assigned room and time, as arranged with your mentor.

Deliverable

Make sure to deliver a copy of your presentation slides to your faculty mentor, in whatever format your mentor prefers, including none.

- Be clearer on who the purpose of the project is for (athletes vs PD patients vs both)
- Talk about why we choose certain hardware over the other (which I thought you did, but it was a comment)
- Make a diagram for the software architecture around slide 8 or 9
- Make the schedule more gantt (spelling?) chart like
- Basically a clearer schedule that's easier to digest
- System overall implies a data flow, make it clearer, possible provide another image.
- Control flow image not very clear (at least without explaining it in depth
    - Keep in mind our audience for UGRADS (not computer scientists)
- Crows feet on ERD need to be substantiated (how many)
- Describe PERT chart better (or don't include it at all) Removed PERT Chart.
- Overall refer to images more since we have them. Don’t treat them as pictures to fill space
- Testing slide

- Explain how big the backups can get (like the wearable has a couple days)
- Test upload over 3G. Its not preferred though. Data plans (we said WI-FI to save data-plan from early on) so we are good.
- Mention “Improve patients quality of life” earlier
- SLide 8, pick a command and walkthrough it
- Slides a bit wordy (particularly 8, 11, 16)
- Lead the purpose into the requirements more near the beginning
- Lots of text - we are good enough to focus more on the pictures.

Ratio of the image with the text.
Perhaps mentioning the layout of the PD shoe, where is the sensors and where are they located
Talk about the size of storing data at each layer. And what control it