



Kinisi



The Apple Watch Form Correction Coach

Team B3

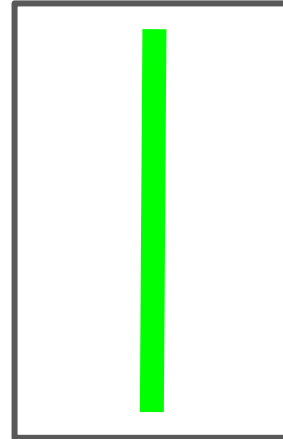
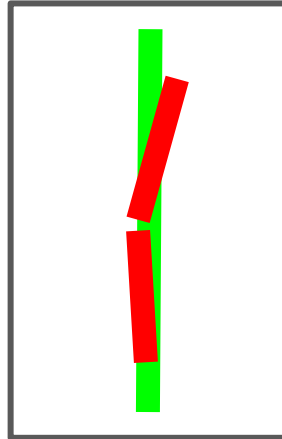
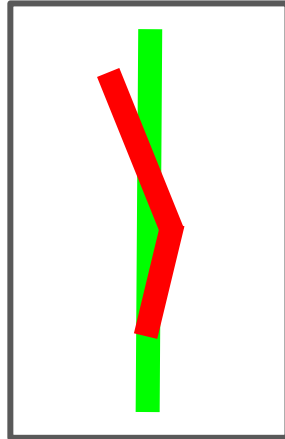
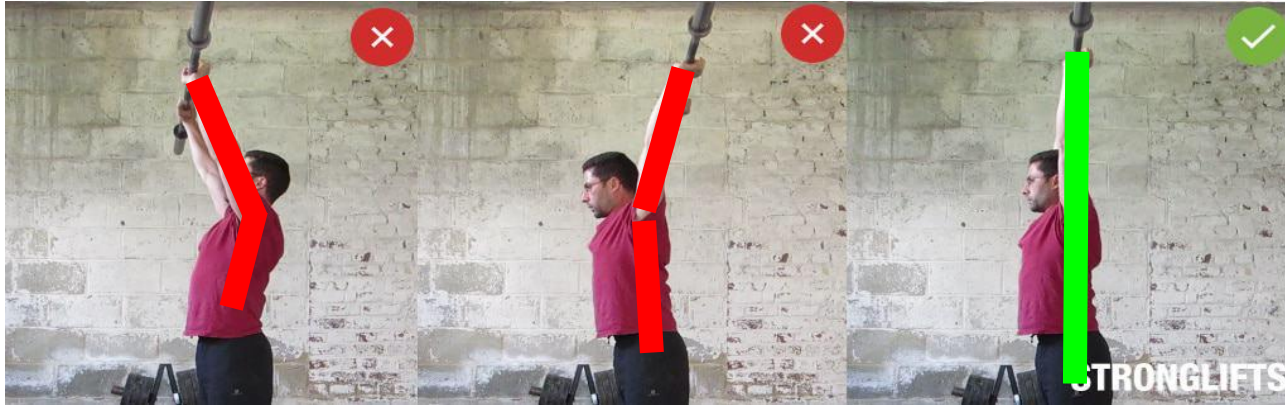
Adrian Markelov and Kyle Jannak-Huang and Matt Spettel

Use Case: Workout Form Correction Coach

- Problem Area:
 - Fitness, consumer tech, coaching, predictive and analytic tech
- Problem Currently:
 - Personal training is prohibitively expensive (\$60+ / hr).
- Training Platform
 - Analyze a user's form in live time
 - Diagnose issues they may have
 - Provide visual and instructional feedback to issue
- ECE Areas:
 - Signal Processing
 - Computer/Software Systems



Final Output: Joint Estimation and Correction



Requirements I

- User Interface - user can easily:
 - Begin and end a coaching session
 - Select a type of exercise to perform
 - Demarcate start and end of sets
 - View form feedback after every set
- Network - Apple Watch ↔ iPhone ↔ EC2
 - Transmitting packets over BTLE (Apple Watch ↔ iPhone)
 - Transmitting packets over HTTP (iPhone ↔ EC2)
 - System handles dropped packets without issues



Requirements II

- Backend Management

- Manage Users: ID, logins, personal training data etc in DB
- Flask: General system management (HTTP, RESTful etc)



- Signal Processing - Data analysis system will be able to:

- Identify demarcations between reps from a set of the desired exercises
- Count the number of reps performed with an average of >95% accuracy
- Process an 'exercise set' of data and recognize form issues with an average >95% accuracy



- User feedback -

- Rendering user's motion, visualizing ideal motion
- Explain issue to user from a set of pre-allocated typical problems (Descriptions are pre-made and stored in DB)

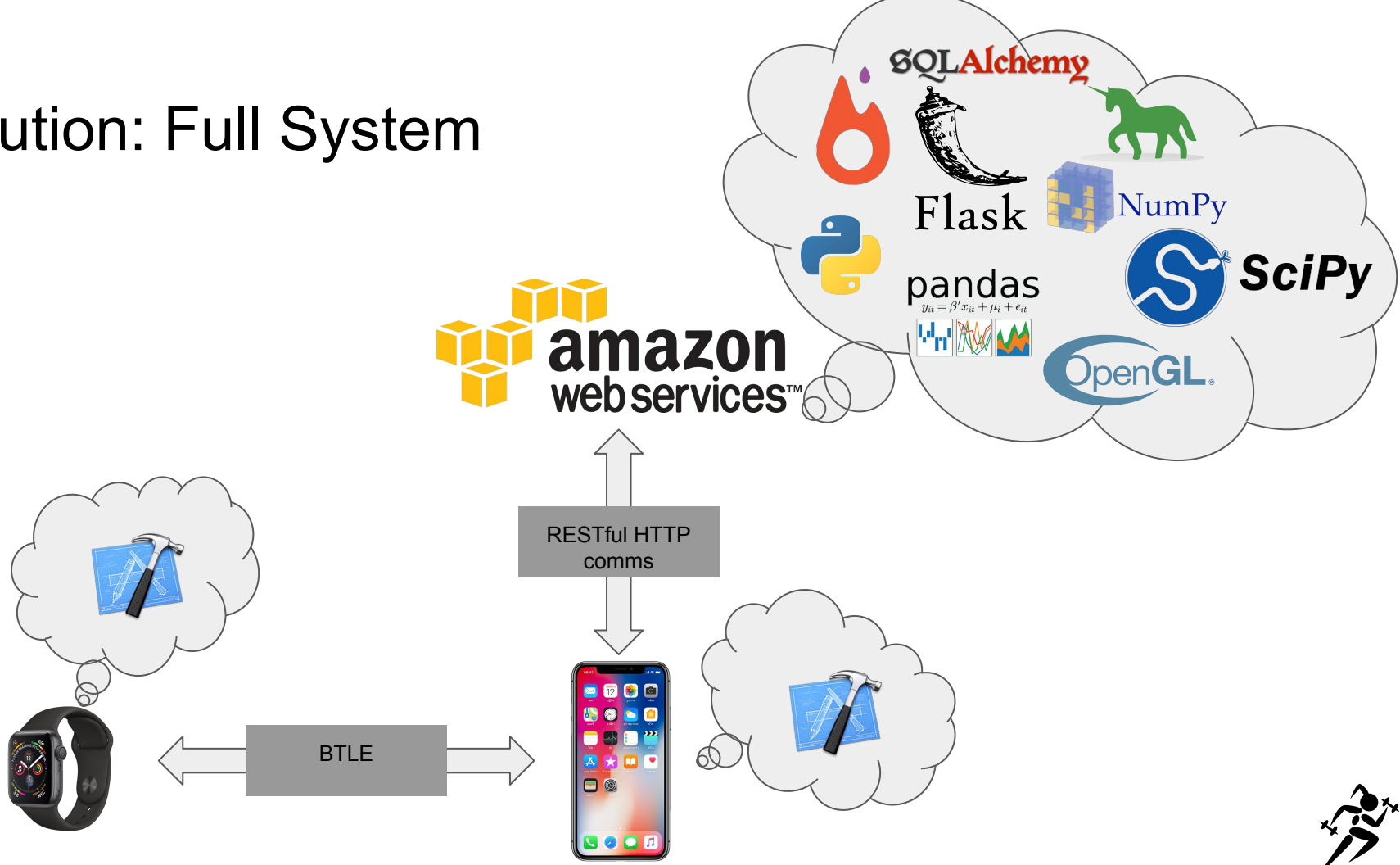


Key Challenges

- **User Interface:**
 - Designing for ease of use
- **Network:**
 - System reliability - no crashes or performance drops
- **Backend Management:**
 - Orchestrating backend systems including: databases, deep nets and graphics generators without clogging the system or crashing
- **Signal Processing:**
 - Demarcating exercise repetitions accurately
 - Training CNN to find faulty form
 - Estimating position of user's limbs from only IMU data



Solution: Full System



Data Processing Solution: Server Side

Backend
Management

SQLAlchemy



Flask



Complete
Backend
Orchestration



Data Analyzing
System



Classify form issue



NumPy



pandas

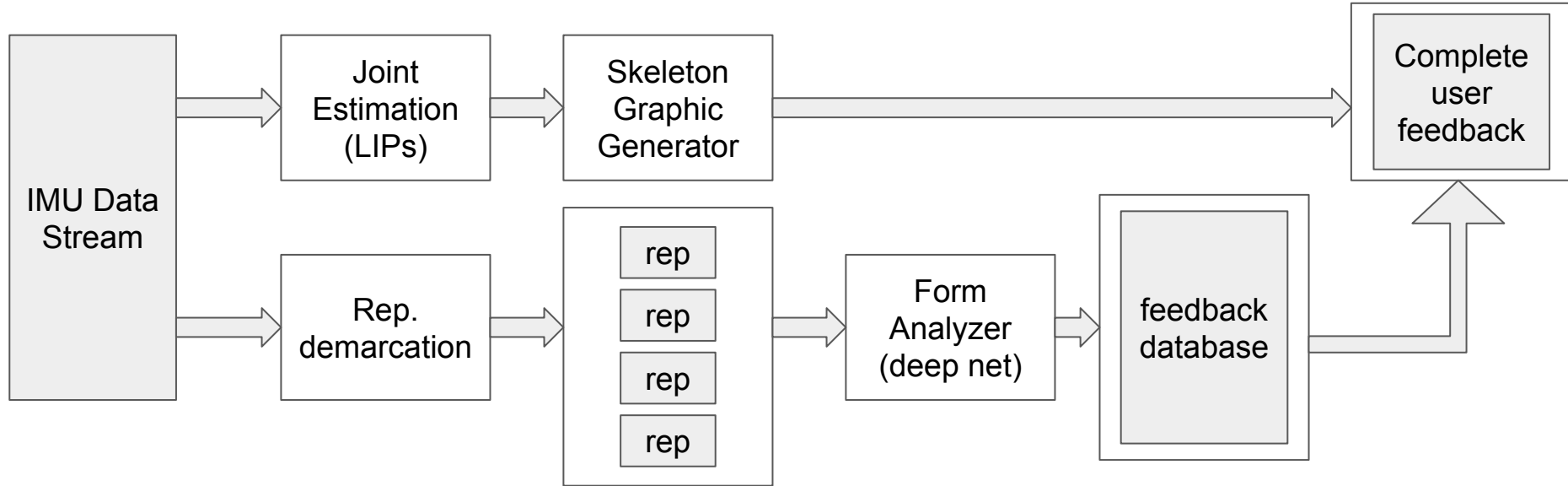
IMU data
preprocessing and
Position Estimation
(inverse problem)



Render Basic
Skeleton Graphic of
good vs bad form



Data Processing Model

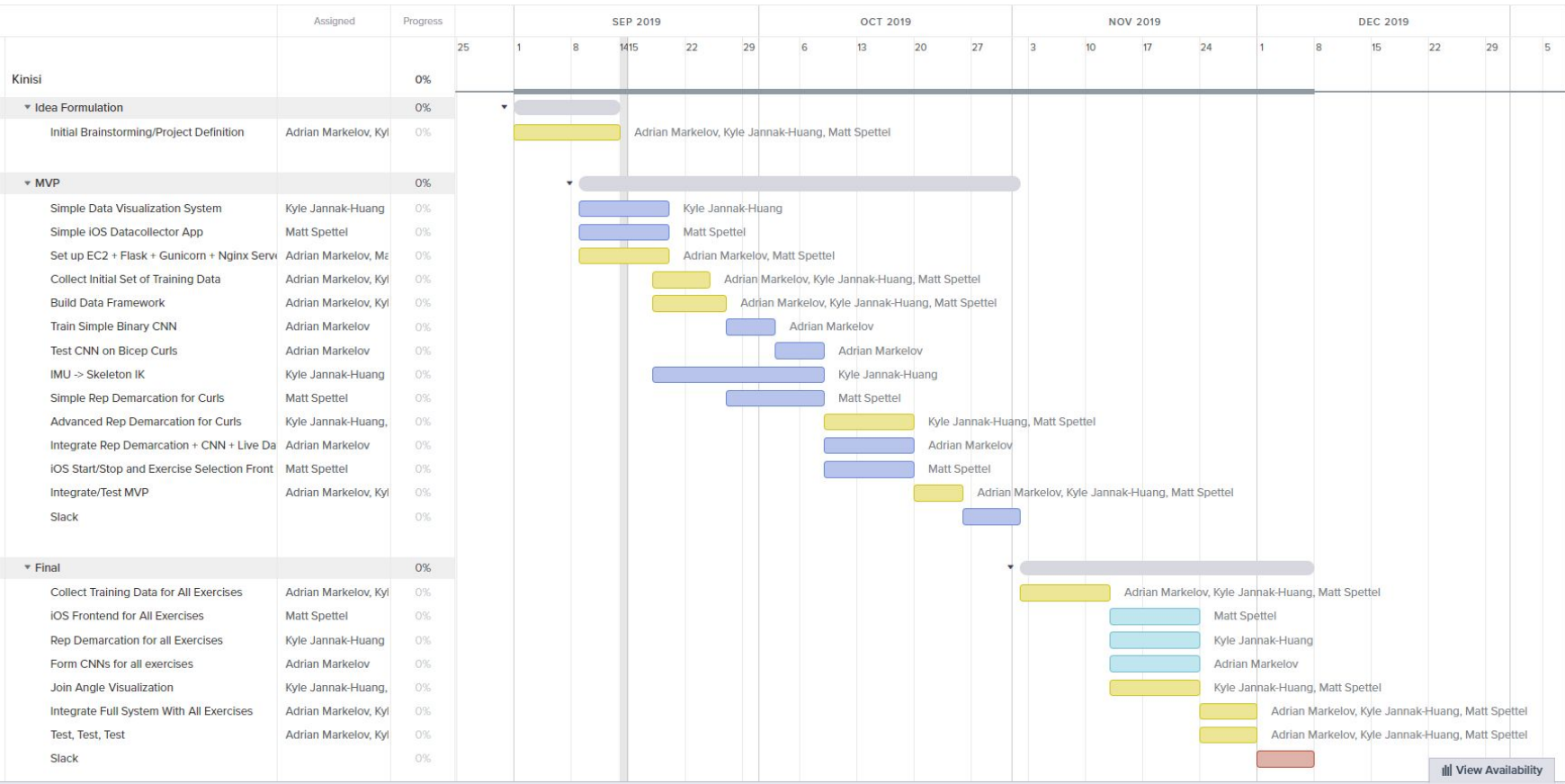


Task Partitions

- Adrian:
 - Backend server management
 - Faulty form detection and classification with CNN
- Kyle:
 - IMU data processing
 - Repetition demarcation
 - User position estimation
 - Graphics generation
- Matt:
 - Backend server management
 - iOS (UI + Networking)
 - Repetition demarcation



Schedule (Gantt Chart)



Testing and Requirement Success Metrics

User Interface:

- If a user can navigate the app and understand the feedback well enough to correct their form without external guidance, the UI is effective.

Rep Demarcation:

- Run demarcation algorithm on every set of training data.
$$\text{accuracy} = 1 - \text{abs}((\text{repsCounted} - \text{totalReps}) / \text{totalReps})$$

Faulty Form Detection

- Run form detection CNN on each rep of training data
$$\text{accuracy} = \text{correctlyLabeledReps} / \text{totalReps}$$

