# PTBuddy

**Reach Normal in the Palm of Your Hand!**
**Graveyard-Spiral ft:**
Shane Grayson, James Nance , Sergio Baray-Miramontes , **and** Michael Garcia **.**

# Project Overview

- PTBuddy is an Android driven application that is designed for  people going through rehabilitation.
- PTBuddy's primary goal is to deliver to the user multiple methods of information on topics of exercise, nutrition, and health benefits during the rehabilitation process.
- PTBuddy's secondary goal is actively track progress of the user and positively reinforce the user when completing a specified task by the program or physician.

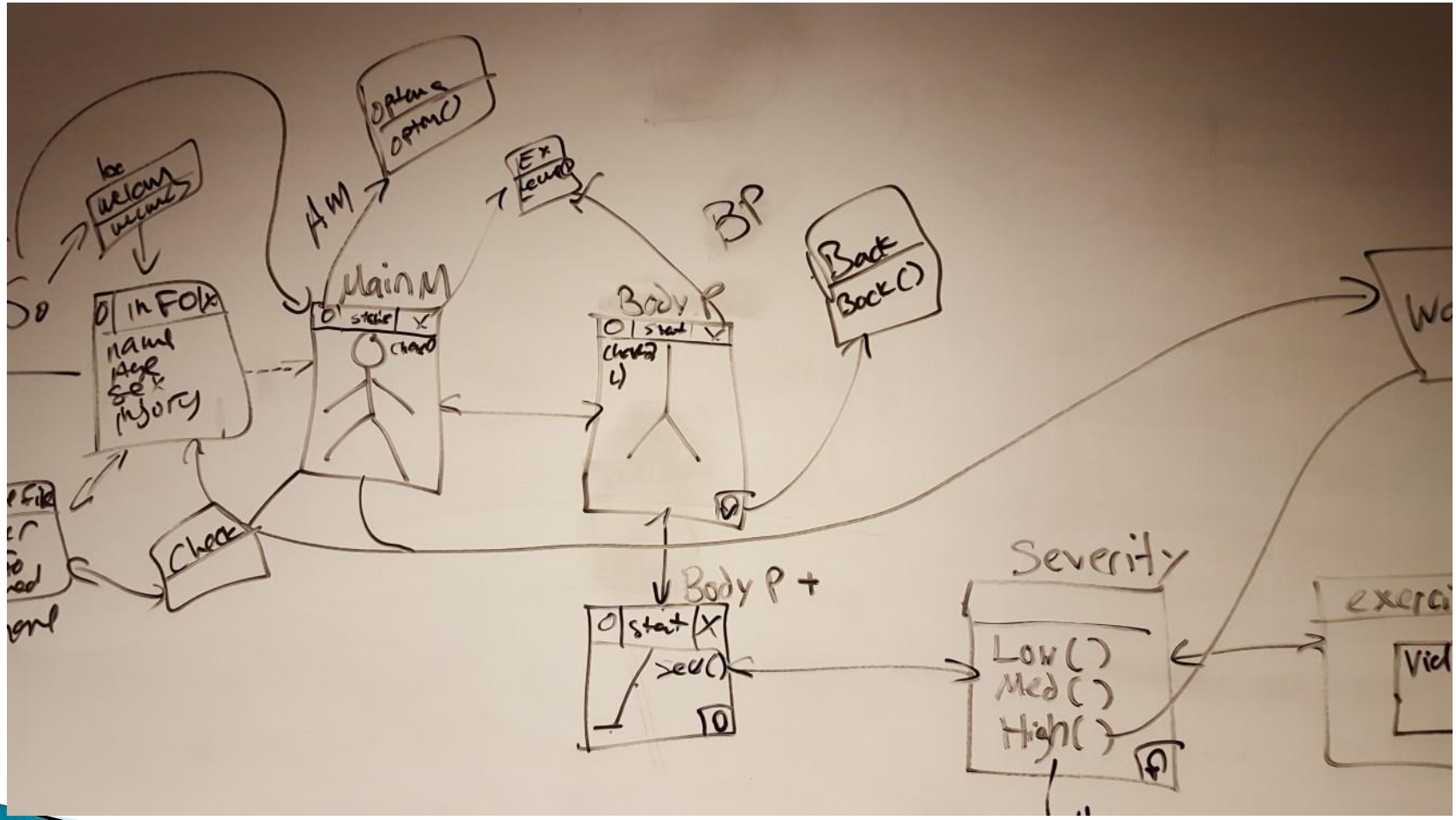
# Key Architectural Drivers

- One major driver was that some of our components rely on information from each other so that ruled out some of the styles. We also wanted something that would be easy to modify since we have some ideas that may be implemented at a later date. Our system is very independent so that is a key driver.

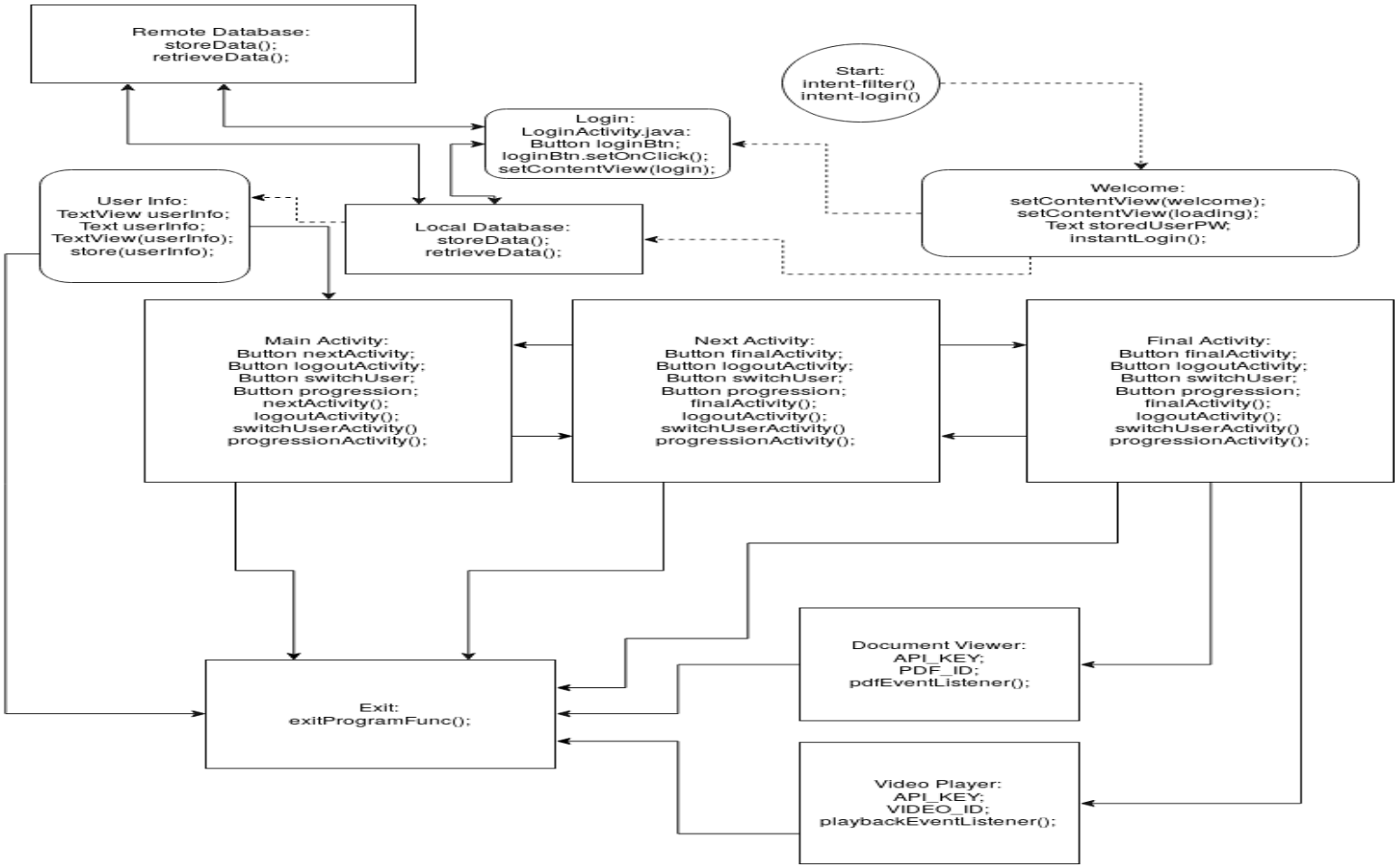
# Architectural Styles

- One style that was considered was the layered system style. The way our system would operate it would be efficient to implement a layered system, for example our exercise lists provide a service to the muscle group. Another style we considered was peer to peer so that the components of our system would act as both client and server so that our components may receive information from each other easily. We decided layered system would work best for our application.

# Old Architectural Diagram



# Architectural Diagram



# Architectural Diagram

- The major focus of our UML diagram is the flow of our program with important functions that are driving it.
- The dotted lines represent a one time occurrence for the user. After they have experienced it will no longer show up for them
- We are currently missing a class in the UML diagram for progression, but that will be implemented in a future demo.

# Conclusion

- Our chosen Architectural style was Layered System Style.
- Some issues we are facing are:
  - personal programming responsibilities, who should be doing what and when
  - Mismatch of development types such, mainly that half of the team are more familiar with Unix based systems, while the other half are more familiar with Windows and Linux based system (Computers on campus do not have Android Studio ☹)
  - Questions? Advice? Traverse activities smoothly? Getting better resolutions from cropped pictures?