# **Tendon Loader Mobile Application Development – Project Summary**

**Project Title:** *Real-time force monitoring for virtual physiotherapy management of injuries – The Tendon Loader*

**Background Information:**

**Achilles tendinopathy (AT) is a common musculoskeletal condition which presents as chronic activity-related soft tissue pain. The tendon injury prevention research group at UBC has demonstrated that Achilles tendon biomechanical properties improve with an exercise protocol that incorporates real-time force monitoring to ensure optimal exercise dose. Rehabilitation of the Achilles tendon using real-time force monitoring could be further enhanced by integrating a home-based Achilles tendon loading device with a Bluetooth-enabled force sensor, which communicates with an application that (1) provides biofeedback and (2) monitors progress towards exercise targets.**

**Project Main Objective(s):**

**This project aims to develop an exercise-monitoring and data-capture system while also providing visually appealing real-time biofeedback to the user via a mobile app interface.**

**Objective: Using established bluetooth-enabled load-monitoring hardware available in the lab, the *Tindeq Progressor* (https://tindeq.com/product/progressor/), develop open-source app software (in Flutter/React Native [or similar] for export to iOS and Android) to display exercise goals to the user, log progress such that a clinician can monitor progress remotely, and also allow for remote tailoring of the exercise program (e.g. change in number of sets, reps, target load, etc.) by a clinician.**

**Project Main Deliverable(s):**

**Fully functional iOS and Android apps capable of interfacing with the *Progressor* and providing basic data visualization/biofeedback to the user. Clinician/research staff should be able to monitor/modify data.**

**The level of testing and functionality should be sufficient to allow for rollout to end-users in a randomized prospective clinical trial assessing clinical status over time in individuals with unilateral AT. Documentation sufficient to allow our lab to evolve the design and refine the device/app interface.**

**Minimum Viable Product (MVP):**

* Mobile app (iOS/Android) which can help guide users through an exercise-therapy regimen by using data visualization and biofeedback (e.g., colour changes, sounds, etc.)
* Visualization of exercise summaries over time (e.g., number of reps completed/day, peak force achieved/day)—exact data to be summarized still TBD
* A way of allowing the clinician to see user summary data + modify exercise prescription (e.g., separate ‘admin’ login account to app? Web portal?)
* A way of exporting the data from a specific user over their entire program (~12wks)—exact data to be exported still TBD, but will need to store data for the 12wk program on the user’s device

**Project Milestones:**

1. **Novel software able to read from/write to Tindeq Progressor device**
2. **Real-time data able to be visualized within novel software in the form of a bar graph (load on y-axis)**
3. **Other app ‘modes’ implemented; able to guide a user through an exercise session (e.g., ~X reps and Y sets)**
4. **Summary data visualization implemented (e.g., reps completed/day on y-axis, date on x-axis)**
5. **Patient/clinician login system established to differentiate users within novel software**
6. **Clinician (i.e., admin) access enabled (either within app or possible web portal) so that clinician can see participant data/progress and modify exercise regimen remotely (e.g., change # reps, # sets, target force)**
7. **Ability to export data from clinician interface to .txt or .csv for analysis by research staff**
8. **Documentation and final project handover**

**Anticipated Timeline (Jan-Apr 2021):**

***Note:* As the software development process is new to me and I do not currently have an adequate grasp on the generalized workflow for this type of project, the timeline noted is speculative and may not be representative of the software design/development process. Refinement is needed. -Kohle Merry**

* **Jan 2021 (start date Jan 4)**
  + **Project orientation/introduction**
  + **Background reading + familiarization with Tindeq Progressor softwares (.py & .pdf explaining connectivity)**
  + **Establish Bluetooth connectivity between Flutter SDK and Tindeq Progressor device (read/write)**
  + **Development of basic wireframe designs**
* **Feb 2021**
  + **Begin data visualization of Progressor load data (i.e., real-time force visualization on basic bar graph, numerical force display within novel software interface)**
  + **Establish scheme for data storage—what data is stored, where, for how long, etc.**
  + **Continued refinement of wireframe designs (UI/UX low-fidelity mockups; interface design)**
  + **Expanding development to include other modes (i.e., exercise mode, MVIC mode)**
* **March 2021**
  + **Continued development of UI/UX and visualization of data through other app modes**
  + **Conceptualization and implementation of user data summary displays (e.g., reps completed/day on y-axis, date on x-axis)**
  + **Begin patient/clinician login system development**
* **April 2021**
  + **.CSV export of data (for later analysis by research staff)**
    - **E.g., for one guided session within ‘exercise mode’, export force & time for full session consisting of ~X reps and Y sets**
  + **Documentation & project wrap-up**
* **May 1, 2021**
  + **End of co-op term for Mitul**

**Current Progress (as of 1/29/2020:**

* **Basic wireframes established within Flutter SDK based on initial designs from Kohle/Alex**
* **Bluetooth connectivity between progressor + flutter established (read/write)**
* **Data interpretation developed to read data from progressor and make meaningful sense of that data (i.e., return force on the load sensor & time of reading)**
* **See attached files for current output and sample code**
  + **output.txt**
    - **Displays a test of nearly 8 seconds, where again I issue commands to start and, stop the measurement after 8 seconds**
  + **test.dart**
    - **sample flutter code; to run it go to https://dartpad.dev/ and paste the content**