Fitness tracking Telegram Bot

Timo Lesterhuis

In de fitnesswereld is het gebruikelijk bij te houden wat je doet. Mensen krijgen schema’s met oefeningen, waarbij precies vermeld wordt hoeveel setjes (sets) en hoeveel herhalingen (reps) van iedere oefening gedaan moeten worden. Veel oefeningen kennen nog een 3e component, namelijk hoeveel gewicht er verplaatst wordt tijdens de oefening. Het is voor krachttrainers, maar ook voor sporters zelf handig om te weten hoe zwaar je een bepaalde oefening kan. Zo kan je over een periode de belasting geleidelijk opbouwen (sets/reps/gewicht oefening omhoog). Tegelijkertijd is het leuk om terug te kijken om te zien hoeveel progressie je al hebt geboekt. Momenteel wordt er door het merendeel van de sporters ineffectief de belasting bij gehouden. Papieren schema’s worden vergeten mee te nemen, en een laptop mee met een Excel sheet is ook onhandig. Er is momenteel dus een tekort in de markt aan toegankelijke manier om fitness-belasting bij te houden.

Een oplossing van dit probleem is om het bijhouden van fitness-belasting mogelijk te maken via de mobiele telefoon. Vrijwel iedereen heeft momenteel een smartphone met internet verbinding, en veel contact (tussen krachttrainer & cliënt, maar ook tussen trainingspartners) gaat via messaging apps zoals ‘WhatsApp’, ‘Facebook Messenger’, & ‘Telegram’. Deze applicatie zal toegankelijk zijn via de messaging app ‘Telegram’.

# Requirements

De onderstaande requirements zijn van toepassing op deze applicatie.

|  |  |  |
| --- | --- | --- |
| Requirement ID | Omschrijving | Prioriteit |
| App.1 | De applicatie draait op een RaspBerry Pi (model 3 B). (of via pythonanywhere | 1 |
| App.2 | De applicatie wordt geschreven in Python 3.X. | 1 |
| App.3 | De applicatie zal gebruik maken van de ‘Telegram Bot API’. Hiervoor wordt gebruik gemaakt van een zogenaamde ‘wrapper’ om aan **App.2** te voldoen. | 1 |
| App.4 | De applicatie maakt gebruik van een sqlite3 database om gegevens in op te slaan. (of SQLAlchemy compatible) | 1 |
| App.5 | De database (**App.4**) moet op termijn periodiek worden geback-upt. | 2 |
| App.6 | Meerdere mensen moeten gebruik kunnen maken van de applicatie (start: N=10). | 1 |
| App.7 | Er moeten meerdere gebruikersprofielen zijn:   * SysAdmin (beheerder van de applicatie) 🡨 toegang tot sql query (beveiligd met password) * Admin * Trainer * User | 2 |
| App.8 | De applicatie moet gebruikt kunnen worden in de Nederlandse taal, maar moet op termijn ook in andere talen beschikbaar zijn. (Commands moeten dan ook vertaald worden.. of alle commands moeten worden uitgeschakeld) | 1/2 |
| App.9 | De broncode van de applicatie wordt gehost op GitHub. | 1 |
| App.10 | gebruikers moeten suggesties kunnen doen / code kunnen bijdragen aan de applicatie. | 2 |
| App.11 | Idealiter moeten er log bestanden worden aangemaakt. (in .txt files of opgeslagen in db onder aparte tabel | 2 |
| App.12 | Een test-set zal gemaakt worden om de functionaliteit van de applicatie in kaart te kunnen brengen. | 2 |
| Data.1 | Data moet idealiter versleuteld / anoniem worden opgeslagen. (**App.4**) 🡪 gebruiker hashen (+ eigen secret key, zodat hash niet nagemaakt kan worden) | 1 |
| Data.2 | Log bestanden (**App.11**) mogen geen volledige persoonsgegevens bevatten. | 1 |
| Func.1 | Gedurende een training moeten de volgende variabelen kunnen worden opgeslagen in een database: 🡪 zie SQL table designs   * Datum * Naam training (eventueel) / ID-code * Naam oefening (per oefening) * Aantal setjes * Aantal herhalingen | 1 |
| Func.2 | Alle variabelen genoemd onder **Func.1** moeten voor de gebruiker aanpasbaar zijn. 🡪 via commands & in combinatie met command line utilities | 1 |
| Func.3 | Het gewicht & aantal herhalingen (**Func.1**) moet per set aanpasbaar zijn. (‘/set’ command, of via inline keyboard? (of combinatie?) | 1 |
| Func.4 | De gebruiker moet de mogelijkheid hebben om gegevens te verwijderen uit de database. | 2 |
| Func.5 | De gebruiker kan zijn schema’s opslaan in een database. | 2 |
| Func.6 | De gebruiker kan tot 6 schema’s onder snelkoppelingen plaatsen voor makkelijke toegang. | 2 |
| Func.7 | De gebruiker kan per oefening het aantal sets dat de voorlaatste keer gedaan is terugvinden. | 1 |
| Func.8 | De gebruiker kan per oefening het aantal herhalingen dat de voorlaatste keer gedaan is terugvinden. | 1 |
| Func.9 | De gebruiker kan per oefening het gewicht dat de voorlaatste keer gebruikt is terugvinden. | 1 |
| Func.10 | De gebruiker kan de oefeningen die de voorlaatste keer uitgevoerd zijn terugvinden. | 1 |
| Func.11 | De gebruiker kan per oefening het zwaarste gewicht dat gebruikt is terugvinden. (1RM) | 1 |
| Func.12 | De gebruiker kan een gewicht als percentage van zijn 1RM berekenen. | 2 |
| Func.13 | Alleen de SysAdmin kan gebruik maken van SQL statements. 🡪 password bij nodig | 1 |
| Opt.1 | De gebruiker kan herinneringen instellen voor trainingen. | 2 |
| Opt.2 | De gebruiker kan een voor gedefinieerd schema (zoals Smolov) inprogrammeren. 🡪 twee mogelijkheden: 1. Vaste combinatie van oefeningen (dan al niet in SuperSet vorm) 2. Vaste combinatie + predictie gewicht op basis van 1rm (zoals bij Smolov) | 2 |
| Opt.3 | De gebruiker kan per oefening een grafiekje laten maken met de gewichten die gebruikt zijn. | 2 |
| Opt.4 | De trainer kan toegang krijgen tot de gegevens van gebruikers, mits deze dit toestaan. | 2 |
| Opt.5 | De gebruiker kan een trainer machtigen om inzage te krijgen in de persoonlijke gegevens. | 2 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Command ID | Beschrijving | Prioriteit |
| /start | Begin gesprek met telegram bot, krijg informatie over basis functionaliteit. | 1 |
| /help | Overzicht van commands & hun beschrijving.   * Optioneel: /help <Command ID> 🡪 specifieke uitleg over command. | 1 |
| /vorige | /vorige <oefeningNaam>: krijg de datum + sets/reps/gewicht van de gevraagde oefening. | 1 |
| /begin | /begin <oefeningNaam>: begin met het bijhouden van een oefening (alleen tijdens een training). | 1 |
| /einde | Beëindig de oefening die momenteel wordt bijgehouden (**/begin**) | 1 |
| /sql | /sql <statement>: voer sql statement uit (alleen voor SysAdmin, komt ook niet voor in **/help** lijstje met comands. | 1 |
| /max | /max <oefeningNaam>: maximale gewicht (+ reps) van de gevraagde oefening. | 2 |
| /beginTraining | Begin een training. Datum wordt vastgelegd.   * Optioneel: /beginTraining <trainingNaam> 🡪 voorgedefinieerde training. | 1 |
| /eindeTraining | Beëindig de huidige training (**/beginTraining**). | 1 |
| /vorigeTraining | Krijg een overzicht van de vorige training (datum, oefeningen, sets/reps/gewicht). | 1 |
| /annuleer | Maakt einde aan huidige status, niks wordt opgeslagen. | 1 |
| /verwijder | Verwijder gehele account | 2 |
| /wijzig | Wijzig gegevens in database (precieze invulling nog niet duidelijk) 🡪 customized keyboards? | 2 |
| /schema | Definieer / wijzig / lees uit /verwijder schema (precieze invulling nog onduidelijk 🡪 customized keyboards?). | 2 |
| /1rm | krijg laatste/beste 1rm waarde(n) (voor gespecificeerde oefening). | 2 |
| /zoek | Zoekmogelijkheden in eigen gegevens database (is misschien deels SQL?). 🡪 geen queries, maar voorgedefinieerde searches in alleen eigen data (dus input sanitized) | 2 |
| /authoriseer | Genereer code waarmee trainer toegang kan krijgen tot je gegevens. (🡪 QR code?) | 2 |
| /grafiek | Krijg een grafiek toegestuurd als afbeelding met de vooruitgang. 🡪 of als html? Is wel fancier! (plotly offline API?) | 2 |
| /gewicht | Houd je gewicht bij. | 1 |
| /hartslag | Houd je (rust) hartslag bij. | 1 |
| /slaap | Houd bij hoeveel uur je slaapt |  |