ReinLife: An Open Source Platform for Machine Learning-Based Mobile Mental Health Study

Boer Zhang[*] Wenyun Wang[*] Fadwa Alaskar[*] Walter Williams[*] Weiwei Pan[*] Zana Bucinca[*]
Finale Doshi-Velez[*]

Abstract

While Mobile Health (mHealth) applications (apps) for personal development are proliferating, current tools do not necessarily help build cross-cutting skills in users. In this paper we introduce ReinLife, an open source mobile app that enables Machine Learning (ML) and clinical researchers to dynamically configure experiments that help users to build new skills. ReinLife allows researchers to design and display surveys where users could self-report their state in the skill learning process. Researchers can then send ML-based personalized interventions to users according to the their state. The design of ReinLife is presented, together with an preliminary analysis on the current version of the app. ReinLife provides a significant step toward the design and implementation of holistic and cross-cutting skill learning mHealth platforms that facilities people's everyday life.

1. Introduction

Mobile health (mHealth) is a growing area within the realm of technology and healthcare. There are plenty of existing health smart-phone delivered applications which help people to improve mental health, gain new skills, and get remote medical consultants (Chan et al., 2015; Clough & Casey, 2015; Kretzschmar et al., 2019). In this current work, we focus on the design of a mHealth platform that allows customized interventions (e.g. personalized exercise suggestions) for cross-cutting skill building.

The design of mHealth interventions, especially when aimed at cross-cutting skill development, demands a multifaceted approach that takes into account the diverse and interconnected needs of the target audience. To systematically explore how interventions can influence the skill development process of users, mHealth platforms that track their reactions on interventions and allow researchers to test different machine learning (ML) algorithms will be extremely helpful. While appearing as a mobile application on the user side, the platform will allow ML and health researchers behind the scene to set up trials and manage user data. Numerous existing mobile health platforms show great potential to (1) provide timely support, (2) ease the costs of mental healthcare, (3) combat stigma in help-seeking, and (4) enhance therapeutic outcomes (Koh et al., 2022). However, to our best knowledge, there are no existing open-source platforms which allows personalized interventions that specifically focus on skill learning processes.

In this research, we present ReinLife, an open source mobile app for ML (especially Reinforcement Learning, RL) based mobile mental health study. ReinLife allows ML and clinical researchers to design and implement cross-cutting skill-learning experiments for users. Researchers can dynamically adjust the experiment information from our Python backend APIs. The researchers can also easily personalize the intervention for each user based on their customized RL algorithms. Our platform enables researchers to skip the substantial time of designing new apps and directly focus on conducting experiments on users' skill-learning behavior, analyzing the data, and testing their machine learning model.

2. Related works: Existing mHealth apps that focus on human's skill-learning process

There are plenty of existing applications with different goals, such as helping ADHD patients to better concentrate by getting more physical activity(Schoenfelder et al., 2017), helping users to reduce stress (Blázquez Martín et al., 2018), alleviating eating disorder symptoms (Anastasiadou et al., 2019), and gain better sleeping patterns (Choi et al., 2018). However, these apps can not be easily adapted study cross-cutting skill learning process because: (1) They are designed for specific purposes. (2) Most of the existing apps do not offer personalized applications. (3) For apps that provide personalized intervention, the ML algorithm cannot always be easily adjusted. Thus more flexible mHealth platforms are necessary for ML researchers to test their models. A recent platform (StudyU; Konigorski et al., 2022; Zenner et al., 2021) allows researchers to define their own skill-learning experiments, and manage user data. Our ReinLife bears a resemblance to their platform in the sense of offering researchers the flexibility to define their experiments. Furthermore, the ReinLife platform allows users to easily send ML-based personalized interventions as push notifications, which is a new feature.

3. Background: Modelling skill-learning behavior with Reinforcement Learning

The skill-learning behavior of human itself can naturally be modelled as a reinforcement learning process. A person behaves as a RL agent and acts based on the reward they get from the enviroment. In that case, Inverse Reinforcement Learning (IRL; Ziebart et al.) is used to infer the underlying policy of the agent, i.e., the behavior of a person. For example Zhou et al. (2020) applied IRL in a mHealth app to infer users' utility function in order to set the optimal daily step goals. A potential caveat of IRL approaches is that the user is assumed to behave optimally, and ignore the behavorial impairments (e.g. myopic planning) of the user. To resolve this issue, Shin et al. (2022) proposed a new perspective to model the user and the app itself as two separate agents, and they intervention of the app can be modelled as the interaction between the two behaviors. The workflow of the ReinLife app is designed under the framework of Shin et al. (2022), thus we briefly discuss the *state*, *action* and *reward* of the human and the app agents.

The human agent According to Shin et al. (2022) the skill learning process can be modelled as a Markov Decision Process.

The *state* of the user is defined as different stages of the skill-learning process. At the initial state, the user is new to the skill-learing process, and at the end state, the user fully masters the skill. There is an extra absorption state ”disengaged”, where the user no longer follows instruction or even deletes the app.

The *action* of the user is to either keep learning, or fail to keep learning. When keeps learning, there will be a chance for the user to make a progress, i.e., enter the next state, or to stay in the same state and makes no progress. When fails to learn, there is a chance for the user to become totally disengaged, or to stay in the same state.

The *reward* of the user is dependent on multiple factors: (1) The burden (negative reward) of taking the action to keep learning. (2) The reward of finally mastering the skill. (3) The influence of disengagement, which can be either positive or negative.

The app agent The behavior of the app can be modelled by a separate RL process.

The *state* the app agent includes both the progress of the user and the action they take (i.e., whether they keep learning).

The *action* of the app is the intervention it send to users. The intervention can help to (1) reduce the burden of the user to keep learning, (2) help the user to better recognize the long-term reward of mastering the skill, (3) help the users to better understand the consequence of disengaging, (4) help the users to increase the chance of making progress.

The *reward* of the app depends only on the behavior of the user. If the user keep learning, the reward is positive, otherwise the reward is negative.

In the following sections, when mentioning RL-related terms, we refer to the app agent.

4. Methods: The architecture and workflow of ReinLife platform

The platform we aim to create should satisfy the following features:

- Back-end provides a set of Python APIs for the researchers to set up experiments, questionnaires, and send personalized interventions as push notifications to each user
- Front-end is written such that it contains main functionalities that researchers can call from the back-end.

We created this architecture in Figure 1 based on these demands.

The front-end and back-end communicates through a cloud database: user information and questionnaire responses are uploaded onto the database as the user 'states' and 'rewards'; researches can retrieve these information from this database and push inventions, serves as 'actions', to users based on their ML algorithm. These intervention would also have a record stored in the database.

5. Implementation of the ReinLife platform

In this section, we discuss the design and implementation we make for each component: front-end, back-end, and database. We will also discuss advantages and limitations of each design choice we made.

5.1. Front-end

5.1.1. App pages

Front-end is implemented in Flutter because it supports multi-platform app and has a big user community. On home page, the app retrieves basic information about the current experiment and displays them. When a user first uses a device, they need to click the first button ”register the user”,

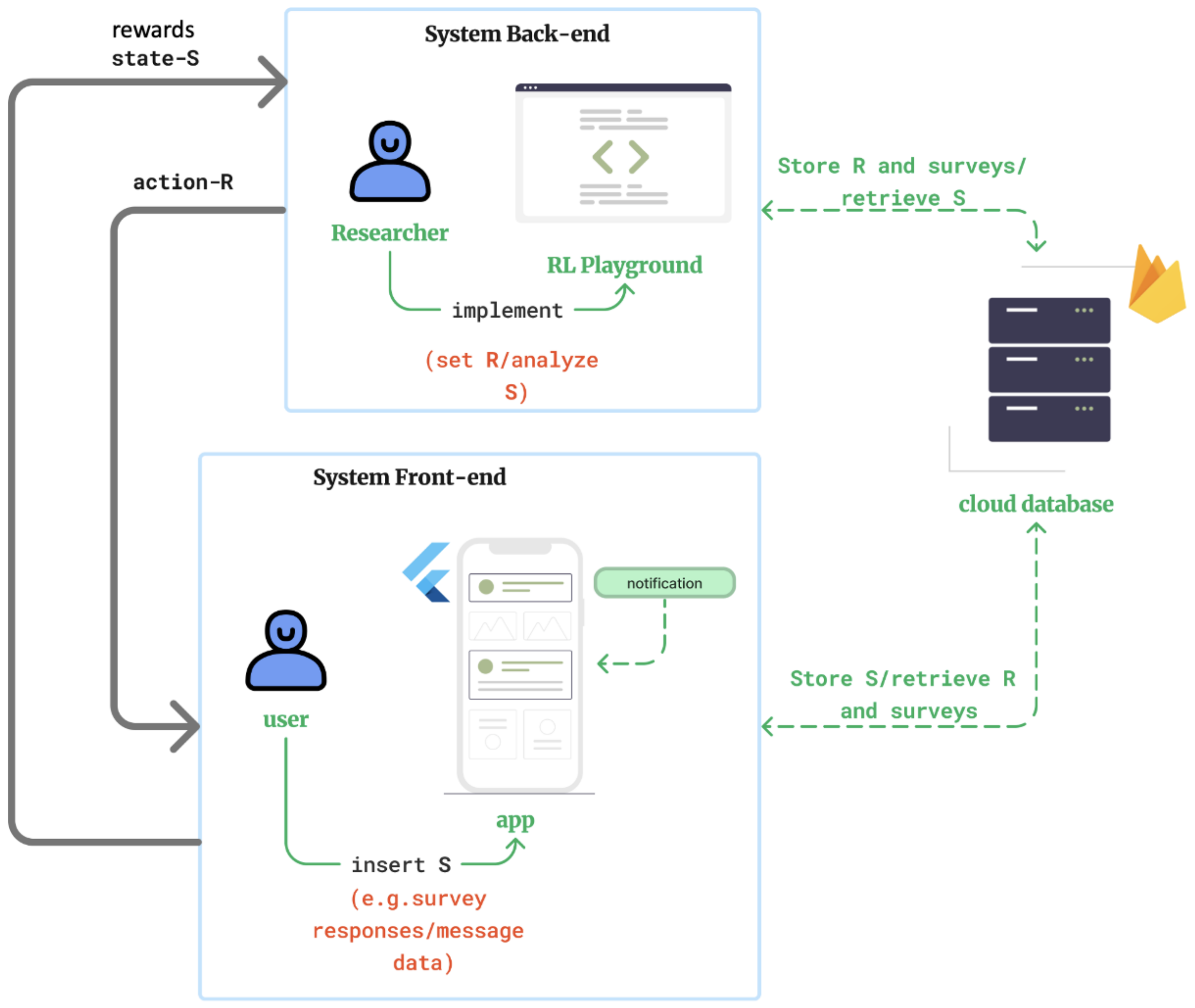


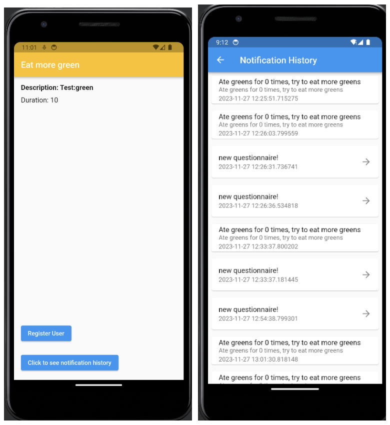
Figure 1. A schematic diagram of the architecture of the ReinLife platform.

so that this token information can be stored under 'Users' collection in the database. There is also a button 'view notification history' on the bottom of the home page. This button navigates users to the full notification history that this user has received in the past, including message type notifications and specific questionnaires they are asked by the researchers to fill out. In this way, user can always fill out the questionnaire or read the message when they have time.

In the front-end design, we decided to use device token as an identifier for the user. This design choice allows users to easily register into the system just by clicking one button and without entering other personal information. The limitation of this design would be that we are assuming each user only access the app from one device all the time.

5.1.2. Notification System

Since the platform uses on reinforcement learning algorithms that rely on the action-state communications between the user and the researcher as a main feature, we designed it to include a push notification-based app that uses notifications as a communication medium to send interventions from the researchers to the users. These notifications are real-time so that the application receives actions/interventions whenever the researchers store them. The app provides two types of notifications: **1) questionnaires**:contain the researcher's surveys (look at Figure 5). **2) messages:** researchers can communicate with and send other interventions to the users as chat messages (look at Figure 4).
Implementation: The notification system is implemented



Description: Test:green
Duration: 10

Ate greens for 0 times, try to eat more greens
Ate greens for 0 times, try to eat more greens
2023-11-27 12:25:51.715275

Ate greens for 0 times, try to eat more greens
Ate greens for 0 times, try to eat more greens
2023-11-27 12:26:03.799559

new questionnaire!
2023-11-27 12:26:31.736741

new questionnaire!
2023-11-27 12:26:36.534818

Ate greens for 0 times, try to eat more greens
Ate greens for 0 times, try to eat more greens
2023-11-27 12:33:37.800202

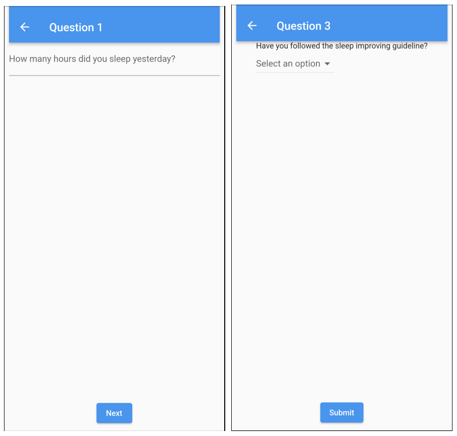
new questionnaire!
2023-11-27 12:33:37.181445

new questionnaire!
2023-11-27 12:54:38.799301

Ate greens for 0 times, try to eat more greens
Ate greens for 0 times, try to eat more greens
2023-11-27 13:01:30.818148

Ate greens for 0 times, try to eat more greens

Figure 2. An example of the homepage (left) and the notification history (right) of the Reinlife app.

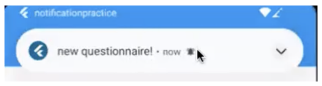


How many hours did you sleep yesterday?

Have you followed the sleep improving guideline?
Select an option

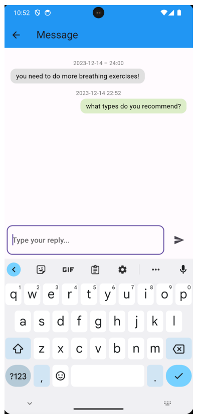
Figure 5. Questionnaire page

by using the cloud database (Firebase) as a tool to store and send researchers' interventions from the back-end to the front-end. The database contains listener functions that are triggered whenever these interventions are stored, and these stored interventions are sent to the users as push notifications once these functions are triggered. We keep a record of these interventions by storing them in the database. Currently, the notification works on Android phones only as activating push notifications on IOS platforms requires Apple Developer Membership, which requires annual subscriptions with high fees.



new questionnaire! · now

Figure 3. A sample notification layout



2023-12-14 – 24:00
you need to do more breathing exercises!
2023-12-14 22:52
what types do you recommend?

Figure 4. Message page

5.2. Back-end

The backend is implemented with Google Firebase service and Python. Firebase Admin SDK is utilized to support the interaction between the Firestore database and the Python codes. In the current implementation, we give the backend researchers full access to the user data. Each researcher is supposed to fork our source code maintain their own copy of the app and the database, in order to ensure data security. Compared to maintaining a centralized backend for different experiments, this design will keep the size of individual database reasonable, which is a more efficient choice. This design will also be a more affordable choice, since Firebase service will give individual users free quota for data storage and running cloud function for the notification system on the GCP. On the user side, we use the Security Rules feature of the Firestore database to guarantee data security, so each user only have access to its own data.

5.2.1. Python APIs

The APIs appears as a Python module 'ReinLifeResearcher.py'. The users can import the module and call the functions to dynamically interact with the users and provide personalized interventions. The APIs allow users to: (1) Set up the information of the experiment displayed in the app; (2) Set up the questionnaires stored in the database; (3) Set the current questionnaire for each user to be answered, and send them a push notification as the reminder; (4) Extract the users' states based on their answer to the questionnaire; (5) Send interventions to users as personalized push notifications, based on their own RL algorithm. The users are supposed to either manually run their algorithm, or design their own automatic program to run the algorithm periodically. We expect the researchers to run the algorithm with the frequency of one or a few times per day.

5.2.2. Firestore cloud database

Our database is shown in the structure shown in Figure 6. We have three collection is total: Experiment Information, Questionnaires, and Users.

In Experiment Information, we have a document named Experiment Information as well, which contains three field: 'Description', 'duration', and 'title' of the experiment that would be displayed on the app home page as a summary of the current experiment.
Then we have the Questionnaires collection, researchers can set customize each questionnaire, so there can be many different questionnaire documents under this collection. In each questionnaire document, its fields includes a 'questionnaireId', which is used to refer to this questionnaire, and a list type 'questions', in which researchers can store many questions of different types. The types of questions we support includes 'Open Text', 'Rating Scale' and 'Multiple Choice'.

In the Users collection, documents are named by the users device tokens. Each token is a document, which contains a field 'token', and two sub-collections: 'notification record' and 'user answers'. 'notification record' keeps a record of all notifications this user has received, including messages and questionnaire reminders. 'user answers' saves all the past answers that this user has responded to each questionnaire.

6. Discussion

Comparing ReinLife with existing mHealth platforms, it stands out as an open-source solution specifically designed for researchers aiming to explore and test personalized interventions in skill-learning processes. The ability to customize experiments and utilize ML algorithms distinguishes ReinLife in the realm of mHealth applications.

While ReinLife demonstrates promise, there are limitations. The assumption of one device per user might not hold in all cases, and addressing this limitation could improve the platform's accessibility. Additionally, user engagement and long-term effectiveness of interventions warrant further investigation through extended studies and user feedback.

The choice of Firestore as the database provides scalability and efficiency, but researchers should be mindful of potential security and privacy concerns. Future iterations of ReinLife could include enhanced security measures to safeguard user data.

In terms of RL modeling, the application of the framework proposed by Shin et al. (2022) adds depth to understanding user behavior in skill-learning processes. However, ongoing research should refine and validate this RL model to ensure its applicability across diverse skill domains.

7. Conclusion

ReinLife represents a novel approach to mHealth platforms by focusing on cross-cutting skill development. The integration of Reinforcement Learning (RL) principles into the design enables personalized interventions, allowing for a more targeted approach to skill learning. The use of device tokens for user identification streamlines the registration process, although considerations for multiple device use should be addressed in future iterations.

In conclusion, ReinLife presents a promising step towards a more comprehensive understanding of cross-cutting skill development in mHealth applications. Its open-source nature encourages collaboration and improvement, and future developments should focus on refining user identification, ensuring privacy, and validating the RL model's efficacy. Through continuous iteration and collaboration, ReinLife has the potential to contribute significantly to the field of mobile health and personalized skill development.

References

Anastasiadou, D., Folkvord, F., Serrano-Troncoso, E., and Lupiañez-Villanueva, F. Mobile health adoption in mental health: User experience of a mobile health app for patients with an eating disorder. *JMIR mHealth and uHealth*, 7(6):e12920, 2019. ISSN 2291-5222. doi: 10.2196/12920.

Blázquez Martín, D., De La Torre, I., Garcia-Zapirain, B., Lopez-Coronado, M., and Rodrigues, J. Managing and controlling stress using mHealth: Systematic search in App Stores. *JMIR mHealth and uHealth*, 6(5):e111, 2018. ISSN 2291-5222. doi: 10.2196/mhealth.8866.

Chan, S., Torous, J., Hinton, L., and Yellowlees, P. Towards a framework for evaluating mobile mental health apps. *Telemedicine and e-Health*, 21(12):1038–1041, 2015. ISSN 1530-5627, 1556-3669. doi: 10.1089/tmj.2015.0002.

Choi, Y. K., Demiris, G., Lin, S.-Y., Iribarren, S. J., Landis, C. A., Thompson, H. J., McCurry, S. M., Heitkemper, M. M., and Ward, T. M. Smartphone applications to support sleep self-management: Review and evaluation. *Journal of Clinical Sleep Medicine*, 14(10):1783–1790, 2018. ISSN 1550-9389, 1550-9397. doi: 10.5664/jcsm.7396.

Clough, B. A. and Casey, L. M. The smart therapist: A look to the future of smartphones and mHealth technologies in psychotherapy. *Professional Psychology: Research*

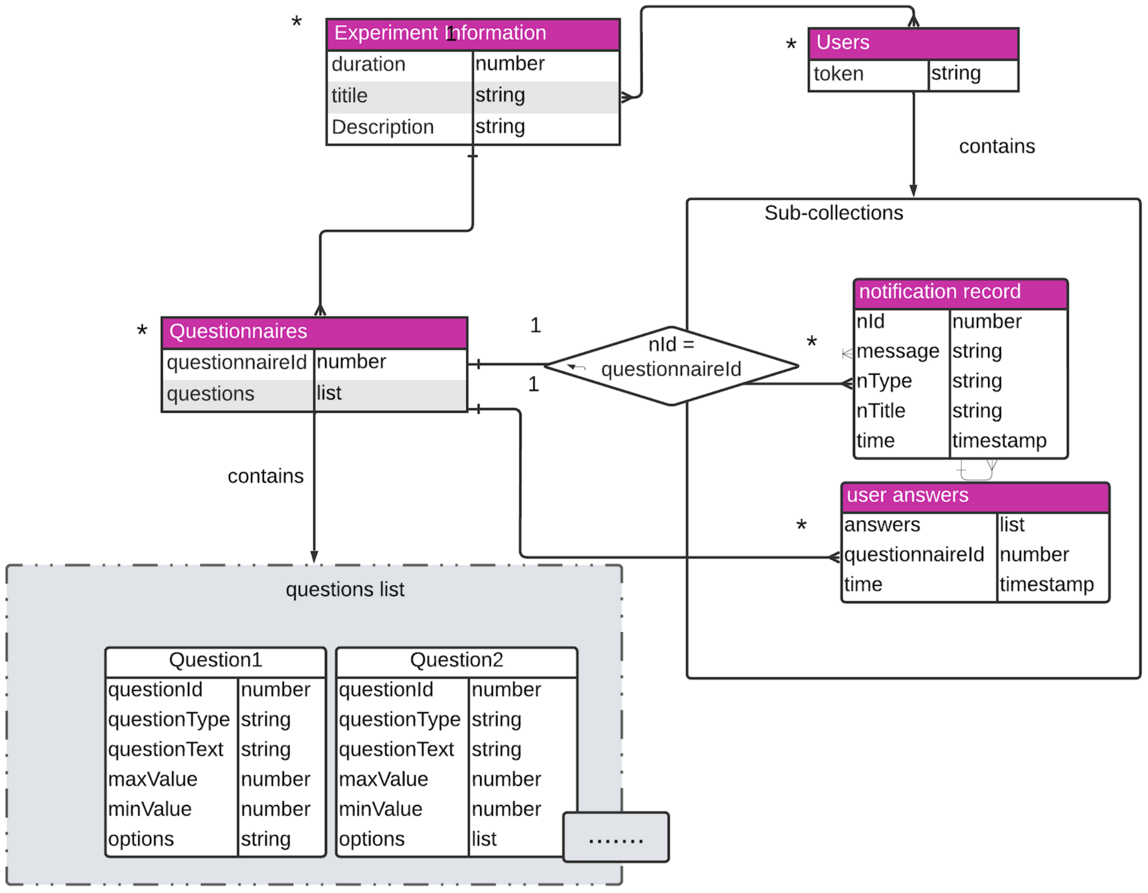


Figure 6. Design structure of Firestore Database.

and Practice, 46(3):147–153, 2015. ISSN 1939-1323, 0735-7028. doi: 10.1037/pro0000011.

Koh, J., Tng, G. Y. Q., and Hartanto, A. Potential and pitfalls of mobile mental health pps in traditional treatment: An umbrella review. *Journal of Personalized Medicine*, 12(9):1376, 2022. ISSN 2075-4426. doi: 10.3390/jpm12091376.

Konigorski, S., Wernicke, S., Slosarek, T., Zenner, A. M., Strelow, N., Ruether, D. F., Henschel, F., Manaswini, M., Pottbäcker, F., Edelman, J. A., Owoyele, B., Danieletto, M., Golden, E., Zweig, M., Nadkarni, G. N., and Böttinger, E. StudyU: A Platform for designing and conducting innovative digital N-of-1 trials. *Journal of Medical Internet Research*, 24(7):e35884, 2022. ISSN 1438-8871. doi: 10.2196/35884.

Kretzschmar, K., Tyroll, H., Pavarini, G., Manzini, A., Singh, I., and NeurOx Young People's Advisory Group. Can your phone be your yherapist? Young people's ethical perspectives on the ise of fully automated conversational agents (Chatbots) in Mental Health Support. *Biomedical Informatics Insights*, 11:117822261982908, 2019. ISSN 1178-2226, 1178-2226. doi: 10.1177/1178222619829083.

Schoenfelder, E., Moreno, M., Wilner, M., Whitlock, K. B., and Mendoza, J. A. Piloting a mobile health intervention to increase physical activity for adolescents with ADHD. *Preventive Medicine Reports*, 6:210–213, 2017. ISSN 22113355. doi: 10.1016/j.pmedr.2017.03.003.

Shin, E., Swaroop, S., Pan, W., Murphy, S., and Doshi-Velez, F. Modeling mobile health users as reinforcement learning agents. 2022. doi: 10.48550/ARXIV.2212.00863.

Zenner, A. M., Böttinger, E., and Konigorski, S. StudyMe: A new mobile app for user-centric N-of-1 trials. 2021. doi: 10.48550/ARXIV.2108.00320.

Zhou, M., Mintz, Y., Fukuoka, Y., Goldberg, K., Flowers, E., Kaminsky, P., Castillejo, A., and Aswani, A. Personalizing mobile fitness apps using reinforcement learning. 2020.

Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning.