

# NFC WorkTracker

A novel approach to put-your-phone-down applications.

Torge Rosendahl

April 13, 2023

**Abstract**—More and more people are having difficulties to focus on their work because they experience distraction coming from their phone, even just by its presence. With this report, I present a novel approach for a work tracking application. NFC WorkTracker motivates the user to leave their phone out of reach when working. This is done by tracking the presence of an application-specific NFC tag in reach of the phone. The app records the times in which the phone is positioned on this tag. The report describes the functionality of the application, the development process, some necessary implementation details and goes into detail on the feedback some test users provided. This project can be seen as a pilot study on the effects such an application can have and lays the foundation for broader research into the effects of motivating users towards not just *not using* their phones, but actually putting them away. This report was written as part of the McGill ECSE 542 — Human-Computer Interaction course.

**Index Terms**—HCI, UCD, Android, NFC, Productivity, Digital Wellbeing, Work Tracker

## I. MOTIVATION

For me as a person, it has always been difficult to “just put my phone down” when I have to get stuff done. And that has not become easier with more social media platforms. It is a known problem that especially younger generations are sometimes struggling to focus on work that needs to be done, due to being used to regularly looking at their phone. More and more young people have difficulties putting their phone away from time to time. What often helped me in such a situation was to put my phone away, ideally in another room, to just not think about having to reach it. But especially the part of putting your phone in another room is often easier said than done. Especially for students that have to study on their own, often at home, this can make the everyday life much more difficult.

Over the last years, many apps have arisen that offer motivation not to use your phone. This is done either through a (soft) lockdown of the device, like with the *OnePlus Zen-Mode*<sup>1</sup> or Android’s *Focus Mode*<sup>2</sup>, or through some kind of gamification<sup>3</sup>, for example in *Forest*<sup>4</sup>. An additional feature most of these apps have is the recording of hours in which the app was active.

For some people, these apps are able to provide a routine of starting their “session” in the app, then working on whatever they have to work on and finally stopping tracking in the

app. For other people, however, it is difficult to develop this routine, because just looking at your phone before starting work might drag them into a notification or some social media app, effectively stopping them from starting to work rather than motivating them. Additionally, they might be waiting for a phone call or want to be able to get some specific kind of notifications. Traditional time-tracking-apps, however, often mute your device, or even completely lock it down for a user-defined period of time to prevent accidental usage.

The act of putting the physical object “phone” down, however, can be much easier correlated with productive time in the brain. Placing the phone out of reach, ideally in another room, increases the ability to focus significantly [1]. And, this also allows you to still hear that important phone call you did not want to miss, because the phone is not locked down and you can hear it ring from wherever it is.

The NFC WorkTracker combines the act of putting down your phone when you start working with the additional functionality of a work tracking application. The presented app will track the time your phone did stay in touch with the surface you put it on. An app that tracks you not only gives you an additional pinch of motivation to not touch your phone mid-work, it also allows additional features, like an automatically stored record of worked hours.

## II. APPLICATION CONCEPT

As the reader might have guessed by this point, the technology used for checking the presence of a *surface* that the phone is laid upon is Near Field Communication (NFC). This requires a secondary device that acts as the surface, in this case provided by a passive NFC tag. Such tags come in the form of actual key tags, plastic cards or as simple stickers. For a general purpose application, stickers are the best choice. The sticker can be placed on the surface where the user wants to have their phone when working.

### A. NFC Usage & Specification

NFC has a maximum communication range of 20 cm [2], which allows for sufficiently easy placement of the phone but guarantees connection loss when the phone is moved. NFC tags can hold any binary data, often stored on the tag as NDEF<sup>5</sup> messages which can hold multiple records. These records can hold a multitude of different things, in my case there will be single message with one record that holds an

<sup>1</sup><https://www.oneplus.com/global/blog/product/zen-mode-explained>

<sup>2</sup><https://blog.google/products/android/android-focus-mode/>

<sup>3</sup>In the example of *Forest*, this includes planting virtual and real trees. Other apps, for example, let you compare yourself with friends.

<sup>4</sup><https://www.forestapp.cc/>

<sup>5</sup>NDEF: NFC Data Exchange Format.

application-defined MIME-type<sup>6</sup> [3] where the content is an ID the identifies the tag<sup>7</sup>.

The data specified on a NFC WorkTracker tag uses the type *application/ca.mcgill.nfcworktracker.tag*, which contains the package name of the project and is therefore application-specific. The contained data is the tag's ID, represented as plain text.

### B. Application Features

The usage of the app itself is very intuitive. Just telling the user (through means yet to be defined) that they can put their phone down onto an NFC tag to track their work hours gives them all the information they need to actively use the system. The NFC tag itself is thereby a valid signifier for where to place the phone, and most people nowadays have used NFC before, for example for card-less payment.

The important part that this app needs to cover in order to be intuitive to use, is to provide the user with all the important information and guide them through the setup process. This setup process consists of buying an NFC tag (or multiple), choosing a place to put it for daily work and then registering it to work with the app. The user will, for the submitted project state, not be guided through the process of picking and purchasing an NFC tag, because any NFC tag works, as long as it is writable. NFC tags come in different shapes and sizes, most often they are found as stickers or in a credit card format. The remaining part of the setup and usage process is guided by the app itself.

The setup process is by far less intuitive than the usage process, so I expected from the beginning that the setup process has the most potential for improvement.

## III. FIRST MOCKUP

### A. Implementation Details and Process

This section will explain the first implementation of the NFC WorkTracker, which is intended to be a proof-of-concept and to be used for the feedback round. This first version of NFC WorkTracker already supports all the required features. This includes tracking of work hours, setting up the application and corresponding NFC tags and viewing the tracking history. I will present these features in the same order I implemented them in, starting with the home screen. This allows me to give some insight into my implementation process along with the presentation of the results.

1) *Home Screen*: The home screen is the landing page when the user starts the app. It has buttons to setup/create a new tag and to show the tracking history.

<sup>6</sup>MIME: Multipurpose Internet Mail Extensions [3]. Used to specify the format of a chunk of data, for example *text/html* for websites.

<sup>7</sup>Actually, the ID is currently neither unique, nor used. The app currently relies on the existence of this data record, which is enough to fulfill the specified use cases.

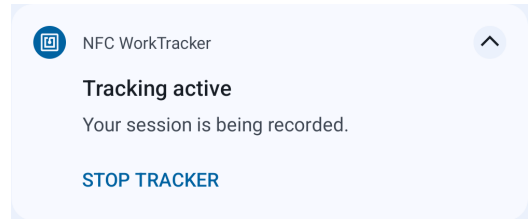


Fig. 1: Screenshot of the “tracking active”-notification. It is shown whenever the tracking service is currently running.

2) *Active Tracking*: The app has to support tracking the presence of an NFC tag in the background. While the user is of course not intended to use their phone during tracking, there is no way to prevent them from, for example, looking at a notification of setting up some other work-related things on their phone, like turning on “Do Not Disturb”. Therefore, within the Android environment, the tracking is implemented as a service. Android differentiates between background and foreground services. The former are used to process things in the background and are not visible to the user. The app should let its users know when a session is being recorded and thus has to be “noticeable by the user”<sup>8</sup>. A foreground service, additionally to being able to access the NFC hardware, also allows the app to show a notification to the user to confirm that tracking has successfully been started. This kind of confirmation is really helpful in building trust into the tracking functionality.

Throughout the implementation, I stumbled upon a minor inconvenience: Android does not allow NFC tags to start foreground services. Therefore, we cannot start our tracking service directly from a tag. The only way to start a foreground service is from another (foreground) activity. Activities in Android are the views of an app that are directly displayed to the user, for example the start screen of an app. Therefore, we have to start an empty activity from the tag and then launch the foreground tracking service from here. The activity is closed immediately after, most of the time the activity is not even displayed to the user because it closes before the open animation is done.

The exact completed usage flow for this tracking functionality is then as follows: First, the user places their phone on the tag. A notification pops up, letting them know that the tag has been detected, see fig. 1. The user then goes on to do their work. When they are done and go to retrieve their phone, they have two means of stopping the session recording. Either, they just lift the phone. When the service is no longer able to access the tag it has been started with, it stops itself and saves the session to the system database. The other option is to open the navigation drawer, selecting the NFC WorkTracker notification (fig. 1) and click “Stop Tracker”.

When the user stops the tracking while the phone is still on the tag, the tag will from then on be ignored. The user has to

<sup>8</sup>“A foreground service performs some operation that is noticeable to the user”, from <https://developer.android.com/guide/components/services>.

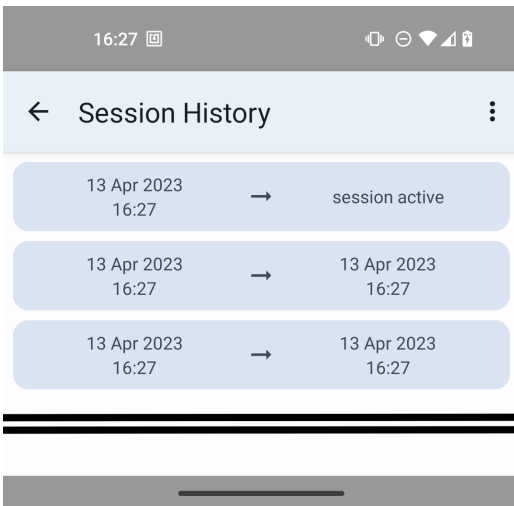


Fig. 2: The history view. Some empty screen in the middle was removed.

lift the phone off the tag and put it back down to re-start the tracking.

3) *History View*: The next section I implemented was the history view. Through this, the user can see the past usage of the tracking functionality. See an example of this in fig. 2.

The history itself is stored in an SQLite database within the application, using Android’s Room abstraction layer<sup>9</sup>. This allows persistent storage between sessions and offers an intuitive way for the developer to access structured data through SQL queries.

The home screen of NFC WorkTracker has a button that launches an activity to display the history. This history-activity acts independently of the rest of the app. For example, showing the history does not interfere with a possible active work tracking process. The history view lists all past history entries. In the current implementation, each entry is represented by its start and end time which is displayed as a date/time combination in the user interface. Displayed times and dates are localized and work across different time zones. However, for privacy reasons, the timezone is not stored with the history entry, so if you move to a different time zone, the times in your history will appear shifted to your **new** time zone. From a design perspective, the app would even work if you were to move timezones during tracking.

In the application bar (the top bar with the back arrow), there is a three dot-menu that has a “clear history” entry. It is currently not possible to delete single history entries.

4) *Setup Process*: Once the base construct of the app was functional, I went on to do the setup process. This process, like the history, is started as a new activity on the press of a button on the home screen of the app.

First, I drew some sketches of what I want the setup process to look like. It is a step-by-step instruction, so a series of consecutive screen makes sense. Only when the current step

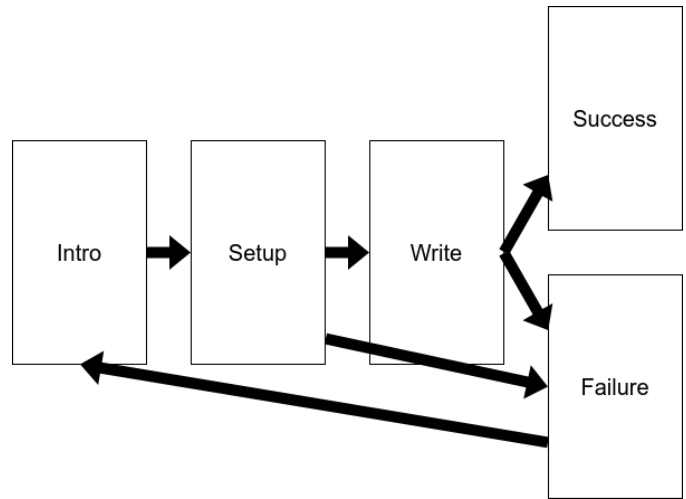


Fig. 3: Simplified version the setup-activity fragment navigation graph.

has been completed, either by the app or by the user, the setup process will display instructions for the next step.

The sketches were done on paper and are not included in this report, but I will explain the functionality I had designed using screenshots from the actual implementation, which are printed in appendix B. The introduction fragment (fig. 5) is used to prepare the phone for the upcoming setup process. It makes sure that the user has NFC turned on and instructs them on how to place the to-be-written NFC tag. The “setup” fragment instructs the user to place the phone on the tag in question (fig. 6) and then waits for that to happen. Then, the tag is written and the user is informed on the outcome, either success (fig. 7) or failure (fig. 8).

Second, I implemented the sketches into Android layout files and added the logic flow. Every screen from the sketches is represented by its own fragment, which represent replaceable sub-layouts in Android activities. For activities that display fragments, Android allows the specification of a navigation graph, to change fragments in a structured manner. A simplified version of that navigation graph is shown in fig. 3. At this point in the implementation, I added temporary buttons to all the fragments that allow me to navigate along the defined actions, to be able to simulate all the read and write interactions with the tag.

Finally, I added all the functionality to the fragments. The fragments most importantly display setup instructions, but some of them have additional logic for interaction with the tag. The intro fragment just statically displays the preparation instructions. The second fragment instructs the user to put their phone down onto an NFC tag. This fragment can only advance to the following step if a tag is detected. Therefore, as soon as it is created, it starts actively searching for tags. For this to work, the automatic (system-wide) NFC detection is disabled in this step. Automatic NFC detection will be reactivated as soon as the setup activity closes (with or without a successful tag write). When this second setup step finds an

<sup>9</sup><https://developer.android.com/training/data-storage/room>

NFC tag nearby, it automatically advances to the write step. This fragment, when it is started, will attempt to write a new NDEF message to the tag, containing a record as defined in section II-A. Since the write process finishes almost instantaneously, this fragment does in reality not show. However, in case the write process does take longer for whatever reason, it still displays a “write in progress”-screen. When the write is completed, the app displays a screen that confirms this success to the user and gives a suggestion on how to proceed. In case the write fails, a failure screen is displayed, with the option to either try again now or quit the setup activity.

### B. Runtime Requirements

To use the app, a small amount of things are required:

- Android phone, with minimum Version 8.0 (Oreo) and has NFC-capable hardware.
- NFC tag, that is capable of communicating with phone (correct NFC technology) and is writable.

## IV. FEEDBACK ROUND

To get some insight into what users think of this first version of the app, I sent out a first copy of the app to two friends. Both testers are students, which matches the discussed target user group. For the purpose of this report we will call them *user A* and *user B*. User A has a somewhat technological background and some knowledge about the inner workings of Android, while user B does not. Both users were instructed to run through the setup process and use the app as far as possible. I created a test instruction document to get comparable results. This document also included a short survey at the end. The following section discusses the results that came up throughout testing.

One thing that came up pretty quickly during testing is that NFC apparently is handled very differently across different android versions and OEMs. While on my testing device everything seemed to work, both users reported different issues with regard to the tracking. For user A, the tracking stopped every time the screen turned off. This is probably related to power saving measures taken by the phone’s manufacturer. User B reported that the tracking did not stop when the phone was lifted. In return, the tracking also did not stop when the was locked. Within the limited time of this project scope, I was unable to come up with a fix for that. There is however a potential solution that will be discussed later. For the sake of this report, I told both user to pretend the app works. User A disabled the automatic screen timeout for the time testing and user B was instructed to stop the tracking manually, through the button on the notification (see fig. 1).

### A. Post-Test Survey

Together with the testing instructions, the testers were sent a post-test questionnaire that they did then send back to me when they were done. A shortened version of this can be found in appendix A. I have restructured the answers into sections.

1) *General Functionality, Bugs and Problems:* As mentioned above, both users had trouble using the app as intended for NFC-related reasons. Those were of course reported upon. As user A reported, some phones do not allow this app to track the presence of the NFC tag while the screen is turned off. Therefore, user A was not able to just put the phone onto the tag but had to take more complicated means to get the tracking to work. This is definitely a large issue that needs to be addressed as soon as possible.

User A, furthermore, reported that it was hard to find a tag to use. I made sure before testing that both users had a tag, but this will of course not be the case for every possible user. It might be required to include instructions for the purchase of (functioning) NFC tags in the app or its documentation. However, I do not like the idea of advertising a specific product or linking an online store that sells tags, not even just “as an example”.

Both users reported that while the testing instructions are certainly useful, the setup process is described well enough within the app that they supposedly would have been able to do it by just using the app.

There was also a problem with the history view. A currently active session is at that point not yet stored in the database and therefore not represented in the history session list. This can be changed quite easily by checking whether or not the tracking service is running when the history is rendered. Additionally, the history needs to be reloaded when a session tracking completes while the history is being displayed. Currently, the user has to close and re-open the history view to update it. Both issues with the history have been fixed between testing and submission of this project.

2) *Branding, Design:* User B reported that they were bothered by the fact that the app includes the word “tracker” in its name, which certainly has a negative connotation. I already feared that this might be the case beforehand, but still chose the name because I could not come up with something significantly better. After all, “work time tracking” is the exact purpose of this app.

Additionally, it was mentioned that the color scheme of the app is not perfect. I took that problem into consideration before, but only implemented it after testing was already completed. The latest version of the app does support Android’s system-wide *Dynamic Color*<sup>10</sup> design scheme.

3) *Feature Requests:* I specifically asked some questions that had the users think about potential new features. I proposed a new feature where you can take notes to document what you did in a tracked session and asked if and how they would use this. Both users agreed that this is a useful feature. However, both users disagreed with my original proposition on how to add these notes. I originally thought that it is sufficient to just have an editable note associated with each history entry, which can be edited through the history view. The testers wanted to have a more direct approach where they

<sup>10</sup>*Dynamic Colors* is also advertised as *Material You*. <https://m3.material.io/styles/color/dynamic-color/overview>

can enter the note during or immediately when stopping the session. No matter the final implementation, the history entry will get *edit*-button to modify said note, which also gives the chance to implement a way to delete just a single history entry.

Another suggested feature was to add more details to the history view. Additionally to the listed entries, it was requested that there be a total for the tracked time, as well as a statistic structured by weeks or months.

A third feature I proposed was to have multiple tags for different things. This plays well with the first feature. A purpose-specific tag could be deployed by allowing a specific tag to set a default note on the created session object. This can be implemented in two ways: either through the tag ID<sup>11</sup> and a local mapping from the tag ID to the default comment or by saving the comment directly on the tag as a string. Both methods have their advantages and disadvantages, it needs some further discussion to figure out the preferred realization.

Finally, one user requested a lockdown of the phone during tracking to force-stop them from using their phone. I will discuss a proposed solution for this in the next section.

## V. DESIGN ITERATION

First and foremost, it has to be noted that asking two users for feedback can not be considered an actual use-case study. I used the feedback round primarily to test the setup-process and the instructions on how the app can be used. From the returned results, it seems as if that was a success and the setup process is more or less good as it is.

### A. Changes based on feedback

The problem that user A reported, where they could not use the app with the screen turned off, is sadly not something I can fix easily. The constraint of disabling the NFC sensor when the screen is turned off is put onto the app by the Android system. The only solution to this is to keep the screen turned on while tracking is active. A functionality like this, however, might light to the opposite of the intended use. By having the screen turned on, the user might actually drawn towards using the system, because it is easily accessible, since you do not even have to turn on the screen to use the phone.

This can be avoided by implementing the soft-lock functionality that one user requested. The app, when tracking starts, could open a black screen (to save as much power as possible) and, while being in the foreground, prevent the screen from being turned off automatically. However, this implementation will also require some additional evaluation with regard to battery usage. Having the phone unlocked permanently will definitely increase power draw.

There might additional alternatives to explore, too. In theory, it is enough to start the timer and lock the phone when the tag is detected for the first time. Then, when the phone is unlocked by the user and the tag is still there, tracking can continue. When the user turns on the screen and the tag is no longer there, tracking is stopped. While this does not guarantee

<sup>11</sup>Remember, this ID is supposed to be unique per tag, even though it is not right now!

presence of the NFC tag during tracking, it is sufficient to be sure that the phone was not used in the mean time.

Then, there were the suggested and requested additional features that can be added to the application. The option of taking notes to a session has been greatly appreciated by both testers. The first implementation for this will just include the chance to add a note to a session from the history view, but the other suggestion will definitely be explored, too. I like the idea of showing a pop-up window to take notes on the current session as soon as it ends. The user is picking up their phone at this exact instance in time anyways. However, for users that do not want to take notes on their sessions, this pop-up will annoy them pretty quickly. I think it makes sense to run a study on this specific feature and the way other users (from a bigger test group) interact with it.

### B. Follow-Up Use Case Study

Finally, I would like to run a bigger use case study, in which I can test the way in which users apply the concepts of this app into their everyday life. By running a bigger study, I might even be able to get measurable results in productivity increase or users' happiness in comparison to not using the app. This study, however, requires a larger user group to average out specific personalities when it comes to phone-usage during working hours. The suggested group size would need to be of at least ten, which is too big to run as a study in this project. Once the app is in a publishable state and all the problems described previously are fixed, it would be interesting to run this follow-up study to actually measure the app's effects. This new test run will also give the option to test other features, like the note-taking pop-up that was mentioned before.

### C. Setup Process Design

With regard to the visual design feedback I got from my test users, I implemented a more consistent color scheme to make the app more visually appealing. However, the setup instructions as they are right now look very improvised. While the text boxes holding the instructions are helpful and also necessary for accessibility, I would like to add some kind of visual guide, as well. For example, I would like to have an image that depicts the process of putting down the phone onto the tag. In the last screen that depicts that the write was a success, a green tick could be displayed to make it immediately understandable that the write was a success. All in all, I think this kind of visual aid would make the process more fun and easy to apply for less technologically experienced users.

## VI. CONCLUSION

In this project, I developed an Android application that allows recording of work time hours based on the physical location of the phone. This incentivizes users to leave their phone in another room when they are working, leading to a more productive working environment with one less distraction.

Doing this project was a great experience and allowed me to get deeper in touch with Human Computer Interaction and User Centered Design. Furthermore, I learned a lot about NFC and the way in which users physically interact with their phone.

The sources of the NFC WorkTracker-App are available at <https://github.com/torgeros/nfc-worktracker>. Additionally, the L<sup>A</sup>T<sub>E</sub>X-sources of this report are provided at <https://github.com/torgeros/nfc-worktracker-report>. This also includes the image files of all the figures in this report.

### Future Perspective

I will transfer this project into the hands of *vastivity*<sup>12</sup> and keep on developing it with a friend of mine. I think this app has potential to be used by a broader range of people, and I will use this platform to push towards bringing NFC WorkTracker into the Android app stores.

### ACKNOWLEDGEMENT

I would like to thank my testers who really supported the development process. Without your constructive feedback this project would not be where it is right now. I would also like to thank Professor Cooperstock for his input into the selection of this project, as well as pointing me into the right direction when it came to this report.

### REFERENCES

- [1] K. Duke, A. Ward, A. Gneezy, and M. Bos, "Having your smartphone nearby takes a toll on your thinking." *Harvard Business Review*, 2018.
- [2] A. Rahul, S. Rao, M. Raghu *et al.*, "Near field communication (nfc) technology: a survey," *International Journal on Cybernetics & Informatics (IJCI)*, vol. 4, no. 2, p. 133, 2015.
- [3] N. Freed and D. N. S. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types," RFC 2046, Nov. 1996. [Online]. Available: <https://www.rfc-editor.org/info/rfc2046>

### APPENDIX A

#### POST-TEST QUESTIONNAIRE ANSWERS

The following is a close-to-original version of the post test questionnaire. I removed a few questions that gave no additional insight and edited the other questions to give some context with regard to the removed parts. Grammatical or spelling errors in users' answers have been copied from their questionnaire.

#### Did you trust the app in setting up everything correctly?

User A: *Yes.*

User B: *I think that overall, the set up went fine. However, I have the impression that my smartphone usage is being observed because of the word "tracking". I do not want the app developer to know what I do on my phone. I know that it is only measuring the time when on the tag but overall I was struggling with its trustworthiness.*

#### What step did you struggle most with?

User A: *Finding a writable tag that I can use.*

User B: *Understanding whether everything worked or not. The notification that the tracking is active did not stop when*

*taking away the phone from the tag so I am not sure whether the app fulfills its purpose yet. I have to end the session by clicking the button in the notification. This needs to be fixed.*

#### Do you feel like you would have been able to set up the app without this instruction document?

User A: *I think I would have been able to do that. It's pretty self-explanatory.*

User B: *Yes, the app is self-explanatory. However, there could be a menu in the app with the steps and suggested using practices so everybody can make the most out of their experience with the app.*

#### How do you think about different session types? Imagine taking notes and setting a color for a specific session through the history menu; or using multiple hardware tags for different type of work.

User A: *Good idea. It could also be useful to take notes for the current session on the home screen of the app. Having tags for different work is a good idea. It could also be useful to be able to filter the history for those different activity types and display summed up time for those activities per week/month.*

User B: *Okay, but having a pop-up where I can directly set a text-tag for the period at hand would be better. I love the idea of different tags.*

#### What other features do you miss?

User A: *Being able to track while the app is in the background/closed. Deleting single items from the tracking history.*

User B: *It can be useful to have the feature that makes it possible starting multiple sessions at once. It would also be nice to include a feature that automatically lock the phone when on the tag and not unlocking it before the next interaction with a tag. It would be nice if the session will also be shown in the app (such as "active") and if the session history updates automatically when ending a session.*

#### Do you have anything else to say?

User A: *The app's color scheme is a bit off.*

User B: *–*

<sup>12</sup>*vastivity* is an organizational level under which a friend and I work on open source projects in our free time. <https://vastivity.github.io>.

## APPENDIX B

### ADDITIONAL SCREENSHOTS FROM THE APP

The following pages hold some screenshots from the app.  
See also the history view, depicted in fig. 2.

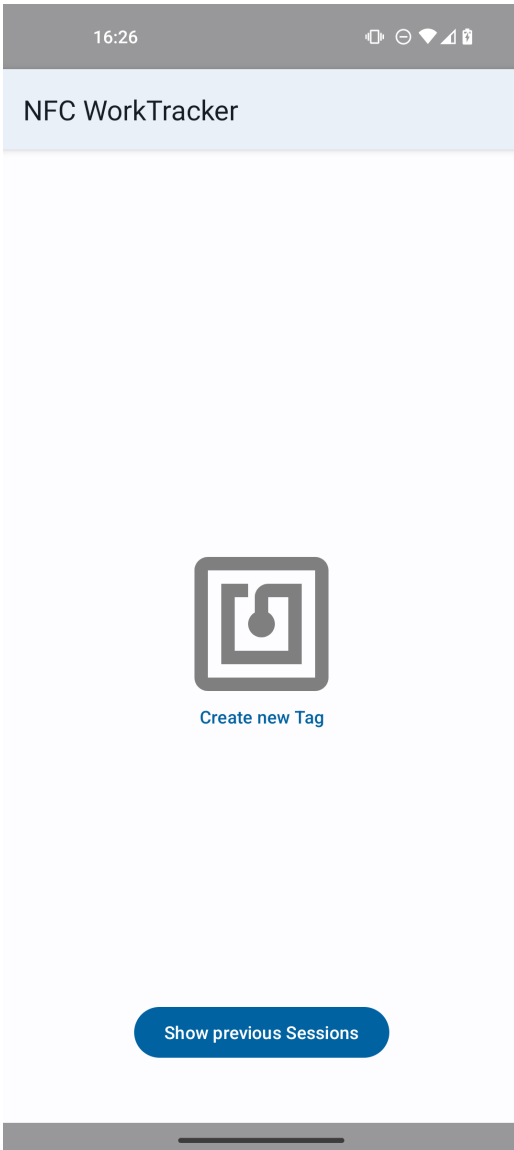


Fig. 4: The home screen of the NFC WorkTracker app.

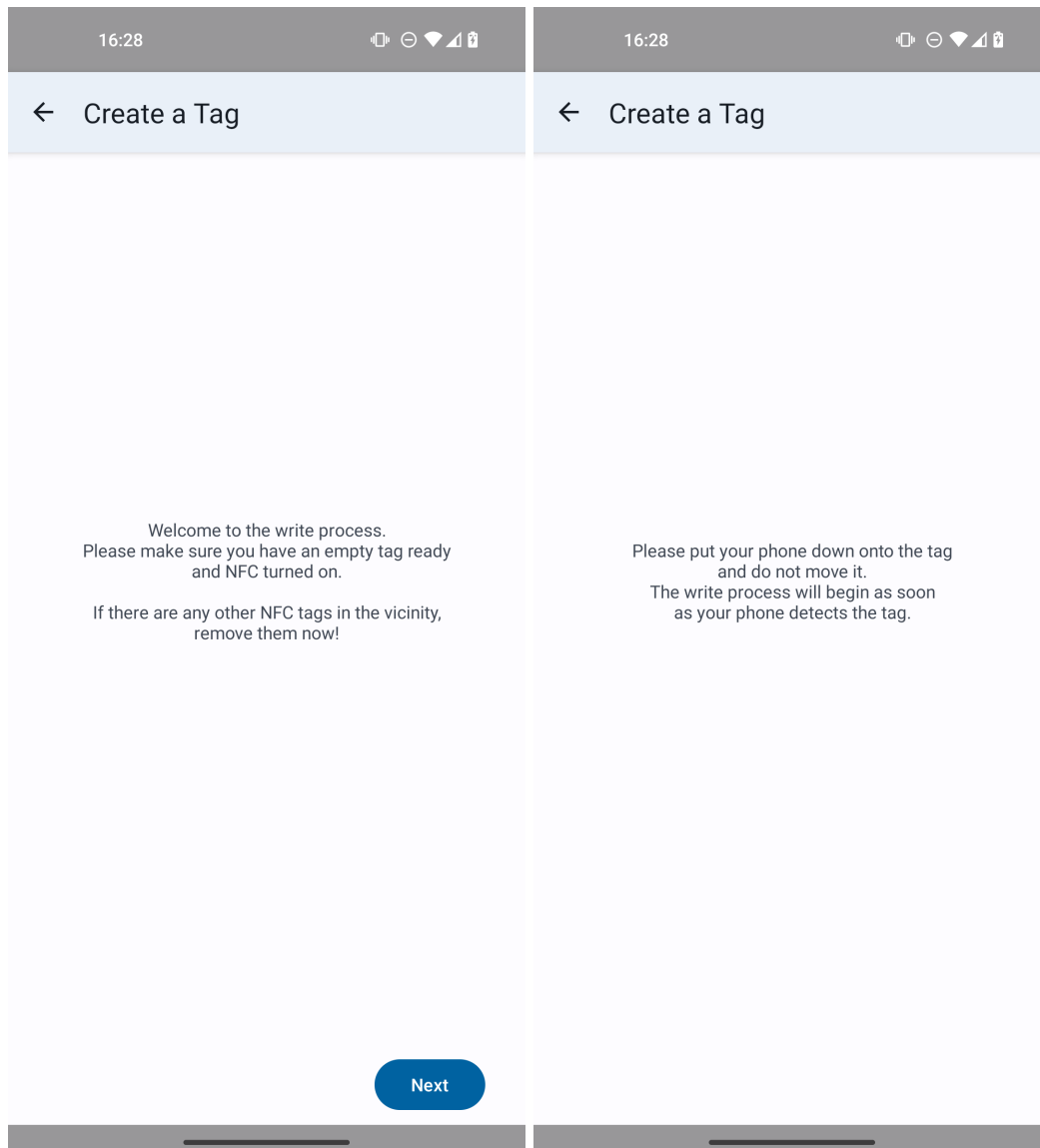


Fig. 5: The introduction screen of the setup process. This corresponds to the Intro-fragment. Fig. 6: The setup screen of the setup process. This corresponds to the Setup-fragment.



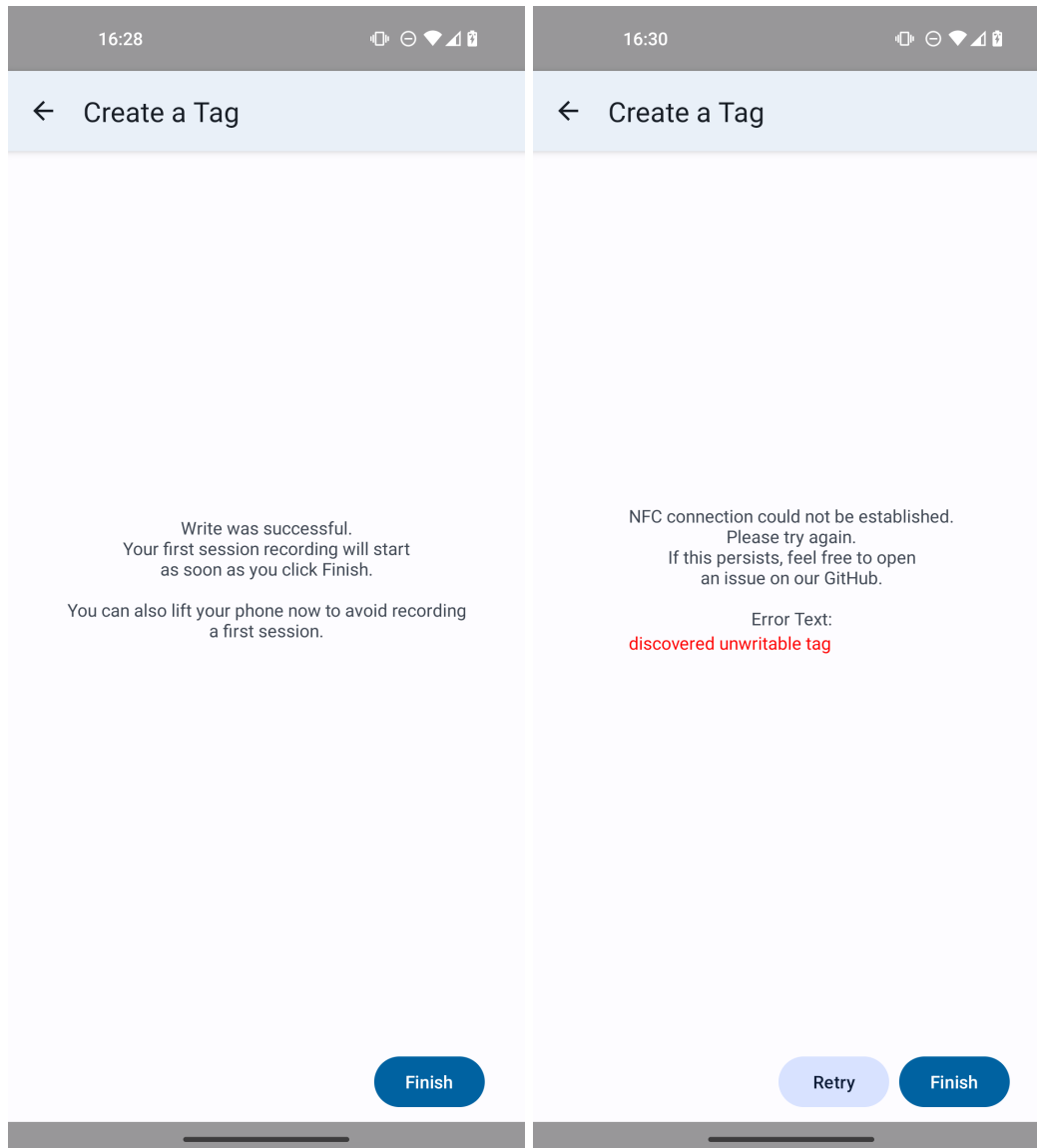


Fig. 7: The write confirmation screen of the setup process. This corresponds to the Success-fragment.

Fig. 8: The failure screen of the setup process. This corresponds to the Failure-fragment. The error message (red text) is dynamically generated.