Università
della
Svizzera
italiana

**Faculty
of Informatics**

Lecture Mobile and Wearable Computing, fall term 2025
Android Programming Project (APP) - Final project report

Group name: MWC

Group members: Luca Di Bello, Luca Beltrami, Marta Šafářová

Instructions

- Please answer the questions reported in the following pages using the foreseen text boxes.
- Do not change the format, font size or any other elements of this template.
- **Be concise and precise.**
- Do not exceed the given limit of maximal number of characters.
- "The app" mentioned in the questions refers to the Android-based application that you develop in the context of the Mobile and Wearable Computing class.
- Once compiled, please upload the document on iCorsi.

The deadline for sending the document is Monday December 15, 2024, 23:59 CEST.
If you have questions: post your question(s) in the iCorsi forum or e-mail ardian.selmonaj@usi.ch and leonardo.alchieri@usi.ch.

1. What is the name of your app?

MindBricks

2. Which problem does the app solve? (Max. 200 characters)

The app helps users stay focused and understand the study conditions that help them stay focused by tracking their study sessions and providing insights based on sensor data.

3. How does your app solve this problem? (Max. 650 characters)

The app uses a Pomodoro timer to structure study sessions and leverage on-device sensors to collect environment signals such as noise (to infer whether the user is in a quiet or distracting place), light conditions (some users focus better in darker settings), and the number of device pickups (to measure distractions). Using data collected from the previous study sessions, the app generates a study plan for the next day that aims to enhance the performance of the student. During each study session, the app offers a gamified experience to motivate users to stay focused. Each session completed successfully contributes to building a virtual city, earning coins and unlocking new building types and customization options.

4. Why is this problem relevant? (Max. 300 characters)

People often study in distracting environments and doní know which factors help them focus best. Combined with constant phone interruptions and weak study routines, staying productive becomes difficult.

5. Do other apps exist to solve this (or a very similar) problem?

☒ Yes
□ No

6. If you answered yes to question 5, list the existing apps that are most related to yours and explain how these solutions differ from your own. If you answered no to question 5, explain why do you think nobody else has solved this problem before. (Max. 650 characters)

Apps like Forest and Focus To-Do offer offer timers and basic gamification, but they do not consider the userś environment and serve only as productivity tools. Our app differentiates itself by collecting sensor data to provide a personalized study plan based on the userś study habits, the environment, and questionnaire feedback. These features are not usually found in existing pomodoro apps, as they usually focus solely on time management and gamification.

7. Which of the built-in sensors of your phone does the app make use of? (Max. 200 char.)

Microphone (noise level), Light sensor (ambient light), Significant Motion Detector (detect device pickups), Accelerometer (kill-switch for light sensor, significant motion detection fallback).

8. Which of the built-in actuators of your phone does the app make use of? (Max. 200 char.)

Speaker (for sound effects), Vibration motor (feedback on session completion)

9. Does your app store sensor data locally, remotely, or in both ways?

☒ Locally ☐ Remotely ☐ Both

10. Motivate your answer to question 9 (i.e., explain why your app stores data and why it does so only locally/remotely or in both ways). (Max. 650 characters)

Environmental and behavioral data (e.g. user habits, information about the student location, questionnaire answers from the user) are sensitive as they relate to mental health and personal habits. The amount of data generated by the app is relatively small and can be efficiently stored and processed locally on modern smartphones. This makes local storage and processing feasible and efficient. Furthermore, user privacy is guaranteed as all data remains on the userś device and is not transmitted to external servers: the user has full control over their data, and can delete it at any time (GDPR compliance).

11. Which type of data visualization your app offers and why. If you answered no to question 11, explain why data visualization is not necessary for your app. (Max. 650 characters)

The app is designed to offer various data visualizations to help users understand and optimize their study habits. These include: (a) **weekly focus chart** (show study minutes and focus score across the week) (b) **hourly distribution chart** (hourly productivity and focus patterns) (c) **quality heatmap** (study quality by day and hour) (d) **Streak calendar** (track consistent study efforts) (e) **daily goal rings** (visualize daily progress towards study time and focus targets) (f) **Recommendation card** (generated optimal daily schedule.
NOTE: We might need to add some visualizations here.

12. Does your app perform any type of data processing on the collected sensor data?

☒ No
☐ Yes

13. If you answered yes to question 12 explain which type of data processing your app performs on the collected data and why. If you answered no to question 12 explain why data processing is not necessary for your app. (Max. 650 characters)

Our app performs time-series analysis on sensor data (noise, light, motion) from study sessions. The core of our processing is a "focus score" algorithm that is intended to quantify user focus. This score would then be used for statistical analysis, pattern recognition, and recommendation generation to help users optimize their study habits. For example, we identify optimal study times by finding the hours with the highest historical focus scores. NOTE: The focus score algorithm is not yet implemented, so these advanced processing features are currently only available with test data.

14. How do you evaluate whether your app performs correctly and achieves its goal (i.e., solves the problem described in question 2)? (Max. 650 characters)

15. Which permissions does your app require to be granted by the user? (Max. 200 characters)

The app requires `RECORD_AUDIO` and `ACTIVITY_RECOGNITION` for core functionality, `POST_NOTIFICATIONS` on Android 13+, VIBRATE for feedback, and `READ_CALENDAR` to support the user's study goals.

16. Does the app raise any ethical issues?

☒ Yes

□ No

17. Motivate your answer to question 16 (Max. 300 char.)

The app's continuous monitoring of user environment and behavior to generate a "productivity score" creates privacy concerns. The continuous surveillance in order to estimate productivity can make users uncomfortable and may lead to misuse of sensitive data if not properly secured.

18. Indicate how each member of the group contributed to the App. Provide the name of the member, his/her main task (UI, database, data processing) and the parts of the code he/she wrote. (Max. 650 characters)

Luca Di Bello (Data Processing & DB): Wrote the sensor drivers (microphone, light, motion), the database logic for storing sensor data, and the focus score estimation. Also handled user onboarding, settings, and data validation. Implemented the current version of the gamification of the app with the city personalization.
Marta Šafářová (UI & Data Visualization): Focused on data visualization and user feedback. She created the perceived productivity questionnaire, implemented the analytics visuals, and developed the interactive study timeline recommendation.
Luca Beltrami (UI & Gamification): Handled the core app experience and gamification. He implemented the Pomodoro timer logic, notifications, sound effects, and started the gamification animations and structure.

19. What are the main challenges you had to cope with in order to build your app and how did you manage to overcome them? Please also describe "failed attempts", i.e., ideas and solutions you might have explored but did not bring to the expected result. (Max. 1 page)

At the beginning, we had difficulties deciding how to manage the sensors, as the app was initially designed to work in the background. However, due to Android permission constraints and sensor usage rules, we decided to run the app in the foreground instead.

Data visualization also proved to be challenging, and we revised it multiple times to make meaningful and effective use of both the sensor data and the user questionnaire. Toward the end of the project, we decided to shift the concept from having the user build a house during study sessions to building a city instead. We found this approach to be more motivating for users and easier to manage, as handling a city with buildings was simpler than managing a house with furniture or complex structures.

Another significant challenge was designing and structuring the city-building system. Managing the city required handling a large amount of game logic, data structures, asset files, and an inventory system, which initially made the codebase difficult to maintain and reason about. To address this, we refactored the game logic by separating it into ten dedicated files, each responsible for a specific aspect of the system (such as city state, buildings, assets, and inventory). This modularization significantly improved code clarity, maintainability, and overall system organization, allowing us to make sense of the complexity and continue development more effectively.

During development, we usually fixed bugs as soon as we noticed them. However, around three quarters of the way through the project, a significant number of bugs had accumulated, and Luca Di Bello had to spend a considerable amount of time focusing exclusively on bug fixing.

Another major challenge was designing and implementing the post-session questionnaire system effectively. We supported our decisions with research and revised the system several times as we found new material. At the same time, we needed to adapt existing questionnaire models to our specific needs without compromising their validity.

Finally, our perspective on the user experience evolved throughout development. Initially, we planned a large onboarding process with multiple fragments explaining the app and requesting all sensor and notification permissions upfront. Following Professor Santini’s suggestions, we rethought this approach and redesigned the onboarding to be more user-friendly, minimizing wasted time at the start of the experience and requesting sensor permissions at runtime instead.