Smart Step Milestone #2 Requirements

This document describes the requirements for the **second development milestone** (Milestone #2) of the **Smart Step** application.

Milestone #2 focuses on introducing the core activity tracking experience and expanding the main app functionality. At this stage, step counting visualization, background tracking behavior, and basic activity analytics are implemented, building on the foundation established in Milestone #1.

The mockups define the overall appearance and expected behavior of the interface. Specific technical decisions (component structure, state handling, animation implementation) are left to the participants, as long as the final result follows the described logic and interaction patterns.

You can find the Smart Step **mockups here**:

> https://www.figma.com/design/aTiGCCavPLtgZNGXcV23H6/Smart-Step?node-id=27-650

Milestone #2 Goal

- Extend the Main Screen with step tracking controls and visual states
- Implement **background step tracking behavior** and related system notifications
- Add **activity metrics** (distance, calories, time) and weekly insights
- Introduce user interactions for **editing steps** and **daily step goals**
- Define navigation updates and supporting dialogs for activity management

Icons

You may use Material Design icons where appropriate. If a suitable Material icon is not available, use the custom icons provided in the Figma mockups.

In Figma, any icon or image can be **exported** by selecting the element and clicking “Export” in the **right-hand panel**. In this panel, you can also choose the desired format (PNG, SVG, etc.).

Adaptive Layouts

The app must support two breakpoints:

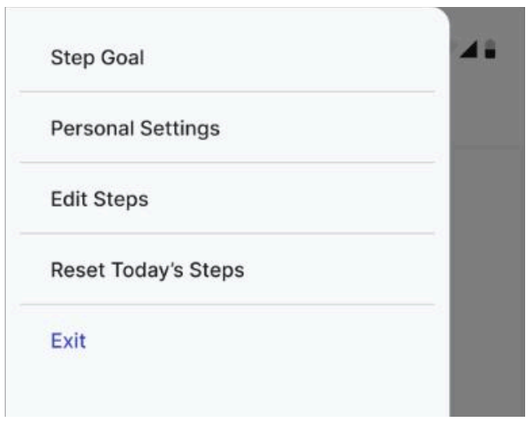
- **Up to 840 dp** → mobile layout.
- **From 840 dp and above** → wide-screen layout.

Technical Requirements

🧭 Navigation Drawer

In Milestone 2, the Navigation Drawer keeps all structure and behavior defined in Milestone 1. The following actions are added.

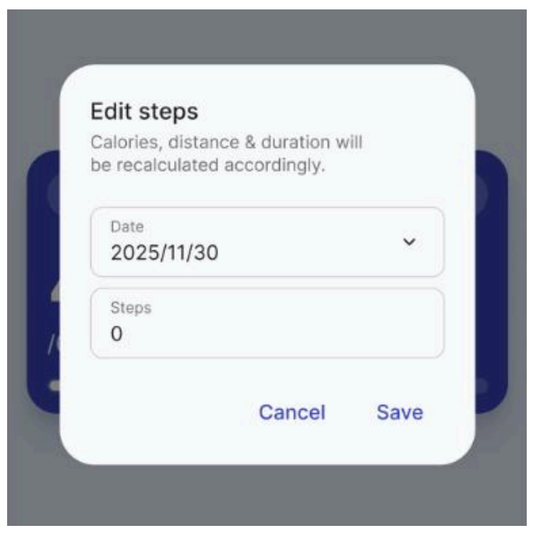
Edit Steps



- Allows the user to manually edit the number of steps for the current day.
- Selecting this item opens the **Edit Steps Dialog**.

Edit Steps Dialog

- Dialog title: *“Edit steps”*.
- Subtitle text: *“Calories, distance & duration will be recalculated accordingly”*.
- The dialog contains two main input fields: **Date, Steps**
- Action buttons at the bottom: **Cancel, Save**

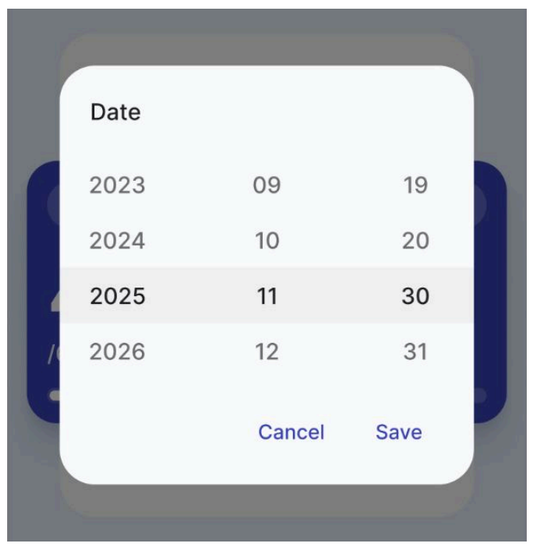


Date Field

- Displays the currently selected date in the format YYYY/MM/DD.
- The field is not editable directly.
- Tapping the field opens the Date Picker Dialog.

Date Picker Dialog

- The Date Picker Dialog is a nested dialog used to select the date for which steps are being edited.
- The Date Picker does not close the Edit Steps Dialog. It temporarily overlays it and returns control back after completion.



Picker Layout and Interaction

- Consists of three independent vertical scroll columns: Year, Month, Day
- Values in each column are changed via vertical scrolling.
- The currently selected value is highlighted in a fixed central selection area.
- Each column scrolls independently.

Date Picker Actions

- Cancel
  - Closes the Date Picker Dialog.
  - Returns the user to the Edit Steps Dialog.
  - The previously selected date remains unchanged.

- Save
  - Confirms the selected date.
  - Closes the Date Picker Dialog.
  - Returns the user to the Edit Steps Dialog.
  - Updates the Date field with the newly selected value.

Steps Field

- The Steps field is a numeric input used to enter the number of steps for the selected date.
- When focused, it opens a numeric keyboard.
- Accepts digits only.
- Does not allow letters, symbols, or decimal values.
- Used to manually define the step count independently of the selected date.

Edit Steps Dialog Actions

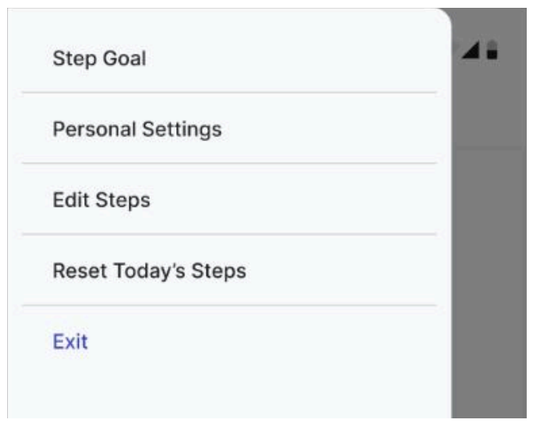
- Cancel
  - Closes the Edit Steps Dialog.
  - Discards all changes made in the dialog.
- Save
  - Saves the selected date and entered step count.
  - Closes the Edit Steps Dialog.
  - Updates the corresponding step data on the Main Screen.

Behavior After Manual Step Editing

When the user manually edits the step count for a day:

- the edited value is saved as the displayed daily step count;
- the current sensor value at the moment of editing is stored as a **new baseline** for further calculations;
- subsequent sensor updates must contribute only **newly detected steps after the edit**;
- the manually set step count must **not be overwritten** by sensor data.

Reset Today’s Steps



The Reset Today’s Steps action is used to reset the step count for the current day.

Since the system step sensor works in a cumulative mode and cannot be reset directly, the app uses a baseline mechanism.

A baseline is a stored reference value of the step sensor at a specific point in time, used to calculate the number of steps for a given day.

Logic

- At the start of a new day or on the first app launch of the day:
  - the current sensor value is saved as the **baseline**.
- The daily step count is calculated as: *currentSensorValue − baseline*.

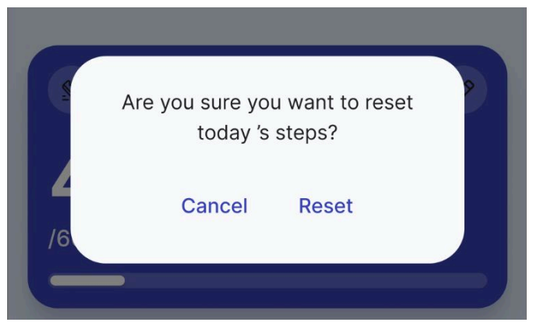
- When **Reset Today’s Steps** is selected:
  - the current sensor value is saved as a **new baseline**,
  - the displayed step count for the current day is reset to 0.

This approach allows the app to correctly reset daily steps without interfering with the system sensor and reflects the behavior commonly used in similar fitness applications.

Reset Today’s Steps — Confirmation Dialog

- The dialog is displayed on top of the Main Screen.
- Confirmation message: *“Are you sure you want to reset today’s steps?”*
- Two action buttons: **Cancel, Reset**

Button Behavior



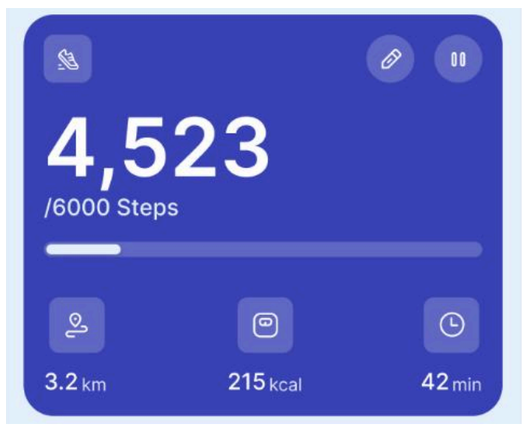
- Cancel
  - Closes the dialog.
  - No changes are applied.
- Reset
  - Confirms the action.
  - Resets the step count for the current day to zero.
  - Closes the dialog.
  - Updates the corresponding values on the Main Screen.

🏠 Main Screen (Updated)

In Milestone 2, the Main Screen keeps all structure and behavior defined in Milestone 1. Only the new and extended elements introduced in this milestone are described below.

Step Counter Card — New Controls

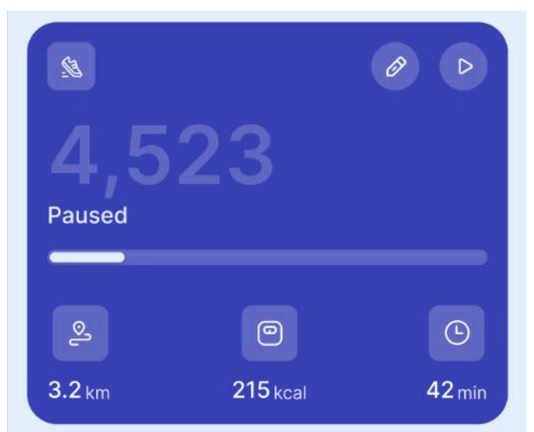
Edit Steps



- An edit icon is added inside the Step Counter Card.
- Allows manual editing of the step count for the current day.
- Tapping the icon opens the previously described **Edit Steps Dialog**.

Pause / Resume

- A control button is added to manage step counting.
- Allows the user to pause and resume step tracking.
- The current state is reflected visually via the button icon.
- When step tracking is paused, the Step Counter Card reflects this state visually:
  - The control button switches from **Pause** to **Play**, indicating that tracking can be resumed.



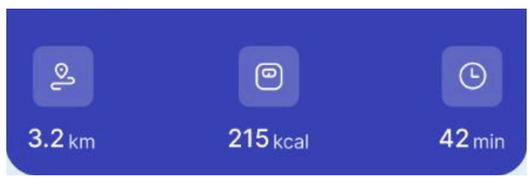
- The step count value appears **dimmed** (reduced opacity) to indicate an inactive state.
- The usual daily goal label below the step count is replaced with the text *“Paused”*.
- While paused:
  - the step count does not increase,
  - no new activity data is recorded until tracking is resumed.

📊 Metrics Section

The Metrics Section displays additional daily activity information below the Step Counter Card. All metrics are informational only and are recalculated automatically based on the current step count.

Each metric consists of:

- an icon representing the metric type,
- a numeric value,
- a measurement unit.



The metrics are not interactive.

Distance

- Displays the total distance covered during the current day.
- The displayed unit depends on the user’s measurement system:
  - **Kilometers (km)** — when metric units are used.
  - **Miles (mi)** — when imperial units are used.
- Distances below one unit are shown as fractional values (e.g. 0.2 km, 0.1 mi).
- The value is rounded to **one decimal place**.

Calculation

- Distance is calculated based on the number of steps and the user’s average step length.
- The average step length is approximated using the user’s height.
- Height is first converted to **centimeters**, regardless of the input unit.
- Step length approximation: stepLength(cm) ≈ height(cm) × 0.415
- Base formula: distance(meters) = steps × stepLength(cm) / 100
- Unit conversion:
  - Metric: distance(km) = distance(meters) / 1000
  - Imperial: distance(mi) = distance(meters) / 1609.34

Calories

- Displays the estimated number of calories burned during the current day.
- The value is displayed in **kilocalories (kcal)**.
- The value is rounded to a **whole number**.

Calculation

- Calories are estimated using a simplified per-step model that takes user parameters into account.
- Formula:
  - kcalPerStep = weight(kg) × 0.0005 × genderFactor
  - calories = steps × kcalPerStep
- Gender factor values:
  - male → 1.0
  - female → 0.9
- If the user's weight is provided in pounds (lbs), it must be converted to kilograms before calorie calculation.

These calculations are approximate and intended for informational purposes only. Medical-grade accuracy is not required for this application.

Time

- Displays the total duration of activity for the current day.
- The value is **always displayed in minutes (min)**.
- If activity duration is less than one minute, **0 min** is shown.
- The value is rounded to the **nearest** whole minute.

Calculation

- Activity time is calculated based on periods when steps are actively detected.
- Seconds are used internally and are not displayed.

📝 Notes

- All metric values are recalculated automatically based on the step count.
- Metrics must be recalculated **only when the step count changes by a meaningful amount**, not on every single step.

Update Rules

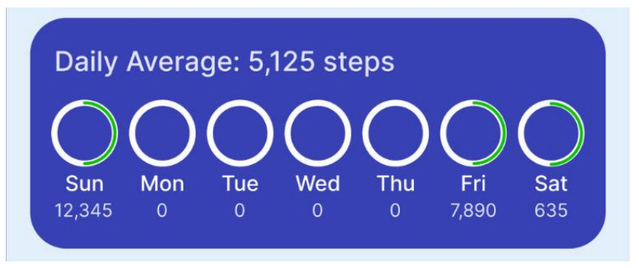
- Metrics should be updated when one of the following events occurs:
  - the step count increases by a **batch of 10 steps or more**,
  - the user manually edits the step count,
  - the user resets today's steps,
  - the app returns to the foreground with updated step data.
- When step tracking is **Paused**, all metric values remain frozen until tracking is resumed.
- This update strategy is intended to balance UI responsiveness and battery efficiency.

📊 Daily Average Section (New)

A lightweight weekly analytics section is added below the main metrics.

Daily Average Value

- Displays a textual summary in the format: *Daily Average: X steps*.
- Represents the **average number of steps over the last 7 calendar days**, including the current day.
- The value is calculated as the arithmetic mean of step counts for all 7 days.
- Days with zero activity are included in the calculation.



Days Row

- Displays a horizontal row of **7 consecutive days**.
- The **rightmost day** represents the current day.
- All other days represent previous days in chronological order from left to right.

Day Item Structure

- a circular progress indicator,
- a short day-of-week label (e.g. Sun, Mon, Tue),
- the total number of steps recorded for that day.

Progress Indicator Behavior

- The circular indicator represents **daily step goal completion** for the corresponding day.
- Progress starts from the **top (12 o'clock position)**.
- Progress fills **clockwise**.
- The fill level is calculated as: daySteps / dailyStepGoal
- Progress is capped at **100%**, even if the daily goal is exceeded.
- If the daily step goal is **changed**, the new value applies only to the **current day and future days**. Progress indicators for past days must not be recalculated.

Steps Value

- Displays the exact number of steps recorded for that day.
- If no activity was recorded:
  - the progress indicator remains empty,
  - the steps value is displayed as **0**.

🔔 Foreground Notification

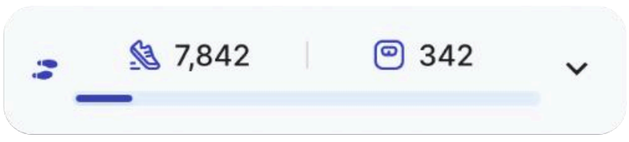
While background step tracking is active, the app displays a system notification in the notification shade. This notification informs the user that the app is running in the background and continues to collect activity data.

The notification has two states: collapsed and expanded.

Collapsed State

In the collapsed state, the notification displays:

- the app icon,
- the current step count,
- the calculated calorie value,
- a horizontal progress indicator.



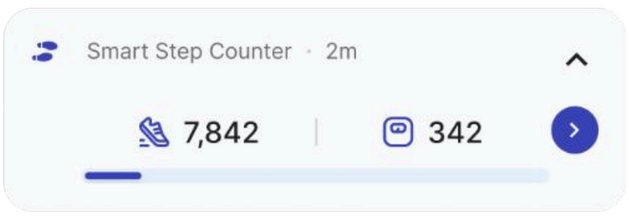
The notification is persistent:

- no sound or vibration is triggered,
- the notification can be dismissed by a swipe gesture,
- if dismissed, it may reappear later when new step data is available,

Expanded State

In the expanded state, the notification additionally displays:

- the app name (e.g. Smart Step Counter),
- a system time indicator (e.g. now, 2m) that represents the time of the l**ast notification update**.



In the expanded state, a chevron-style button is also shown. This button is a **decorative element** and visually indicates that the notification can be used to return to the app.

Interaction

- Tapping the notification in any state opens the app.

Data Updates

- Step and calorie values are updated silently while background tracking is active.
- Updates do not trigger new notifications or draw additional user attention.
- The notification remains visible as long as background step tracking is active, even if the app UI is closed.

Permission Dependency

- The notification is shown only when background step tracking is allowed and actively running.
- If background access is not granted:
  - the background service is not started,
  - the notification is not displayed,
  - steps are counted only while the app is in the foreground.

> 💡 **Technical Note (Android 13+)**
>
> If the app targets API 33 or higher, the POST_NOTIFICATIONS runtime permission must be requested in order to display the foreground notification. This permission should be requested using the standard system dialog. No additional custom UI flow is required.

🔗 Useful Links for This Challenge

- Foreground services overview
- Foreground Services video
- About notifications
- Guide to app architecture
- Create a Splash Screen
- UX With Material3
- Full Guide to Material3 Theming
- Request runtime permissions
- Bottom sheets
- DataStore
- Physical Activity Permission
- Ignore Battery Optimization
- Save data in a local database using Room
- The Full Jetpack Compose Responsive UI Crash Course
- How to Save & Restore the Scroll Position of a LazyColumn Persistently
- Stateful vs. Stateless Composables
- State Hoisting in Compose
- Managing State in Jetpack Compose (Codelab)