GitHub Username: juanmeanwhile

Ringmind me

Description

A common problem for women using ring birth control method is that they have to follow a really specific timing which can be hard to follow without some extra help. There are some apps out there but all of them feel old and their UX can be improved. Ring reminder is simple, useful and nice looking, offering reminders so women don't miss any action to be done.

Intended User

Women using ring birth control.

Features

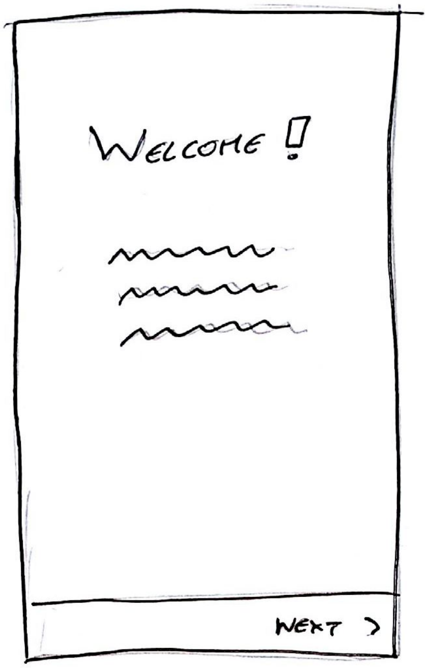
The app main features are:

- Notify about a needed action (put or take the ring)
- Simple UI to watch, from a first sight, next action to be done and when

- Calendar with future actions
- Configurable alarm

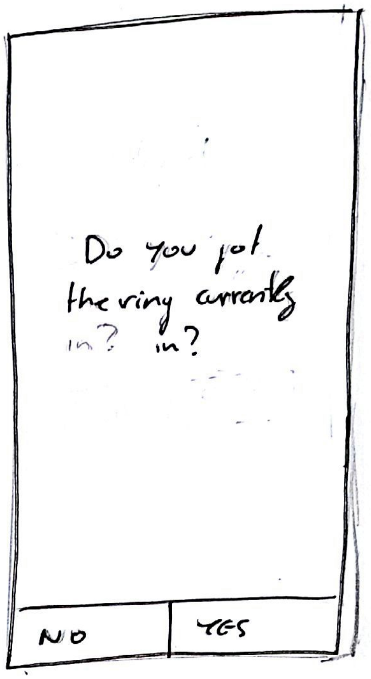
User Interface Mocks

Screen 1 - Welcome



First screen the user will see the first time she's opening the app. Provides an small introduction to the app and has a next button to jump into the next screen.

Screen 2 - Initial setup 1

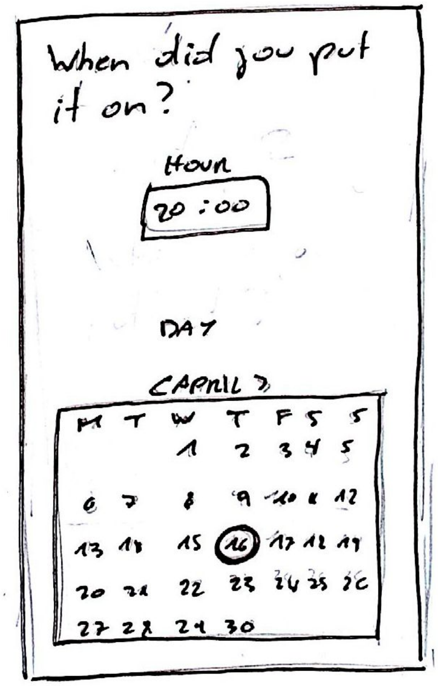


The app will need an initial setup process based on two screens. The user need to indicate the actual status of the “treatment”, which is, if she has the ring currently in or not and when she put the ring in (or take it off).

In this screen, the user is asked about it she has the ring currently in or not. Two buttons allows to select the response.

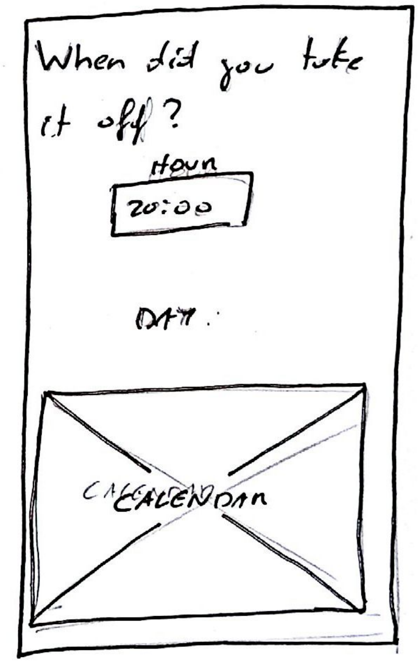
Selecting Yes will lead to screen 3. Selecting No will lead to screen 4.

Screen 3 - Initial setup 2a (datetime selector)



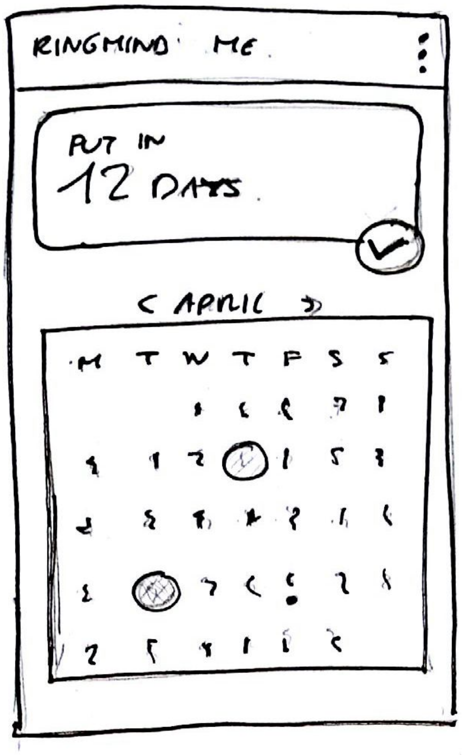
The user can select an hour and a day. Once the user has selected both, she will go to the main screen (screen 5).

Screen 4 - Initial setup 2b (datetime selector)



The user can select an hour and a day. Once the user has selected both, she will go to the main screen.

Screen 5 - Main Screen



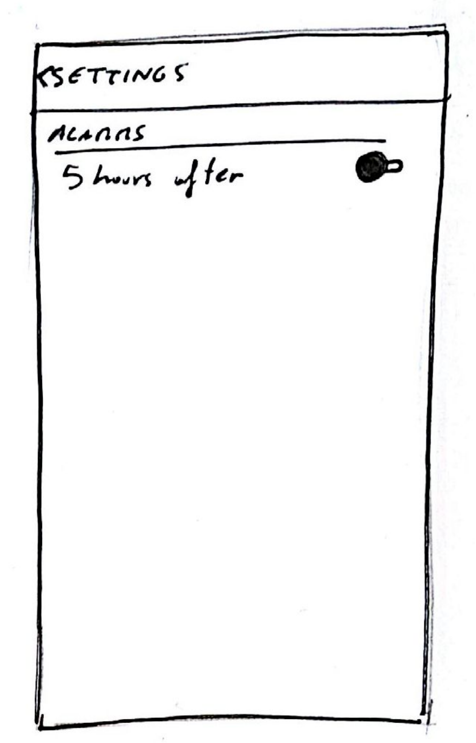
Once the user has done the initial setup, this will be the screen to be shown. When the user enter the app (and the initial setup has been done) this will be the screen that will be shown.

In the upper part it is specified the next action to be done by the user and when. It will be specified in days unless that less that one day is left which will be specified in hours. When the time to perform the action arrives, the circle button will reveal so the user can mark the action as done.

When pressing the button, it will lead to the date and time selector screen (same as screens 2 and 3), so the user can select the time and date when she performed the action, so the next action can be calculated.

Bottom part shows a calendar with all the future actions to be done. The calendar appears in the current month, but user can navigate to other months.
Navigation bar menú button will allow to enter the Settings of the app.

Screen 6 - Settings



For this first version, alarms can be enabled or disabled.

Key Considerations

How will your app handle data persistence?

The app will have a Content provider which will store the future actions to be done and when. This content provider will be used to set the corresponding alarms and to show future and past actions in the calendar.

Describe any corner cases in the UX.

App will show a notification (if those are not disabled) when the user need to perform an action (put or take). When tapping the notification, user will be lead to Main screen (screen 5).

Describe any libraries you'll be using and share your reasoning for including them.

SublimeCalendarView
For the calendar view
(https://android-arsenal.com/details/1/1953)

Next Steps: Required Tasks

Task 1: Project Setup

Create a new project in Android Studio and add SublimeCalendarView library

Task 2: Implement UI for Each Activity and Fragment

Implement the activities and fragments of the app.

- Build UI for MainActivity
- Build UI for IntroActivity
    - Build welcome fragment UI
    - Build date and day picker fragment UI
        - Defines a listener to be notified when the user selects hour and date
- Build UI for Select Date and day Activity (will reuse data and day picker fragment)
    - Will return the selected time and date
- Build Settings Activity

Task 3: Create content provider

Implement the content provider and perform the follow actions in the respective screens:

- Insert next action to be done at the Setup process
    - When an action has been inserted, future actions have to be updated according with the entered time and date.
- Show next action to be done at Main screen
- Show actions to be done in the calendar.
    - When moving to a new month, query the provider to get the actions for the selected month

Task 4: Configure the AlarmManager

The alarm manager will be responsible from showing a notification to the user when an action has to be done.

- The AlarmManager have to be configured as a result of the setup process
- Configure alarmManager at system startup.
- AlarmManager will query the provider to get the info about when the alarm has to be configured.

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”