

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: [Rafabox001](#)

BeOnTime

Description

In this world the most valuable thing for many people is time, time cannot be bought and it is irreplaceable. So being always in time for every task that you plan or schedule throughout your day becomes of high importance.

BeOnTime is an app that addresses this issue by letting us create not just an alarm, but a personalized set of tasks that we need to complete before achieving the desired goal "ON TIME".

For example if I want to get at my work at 8 am. I need to consider all my tasks normally done before going to work, for example:

Waking up (10 minutes)

Take a shower (20 minutes)

Walk the dog (30 minutes)

Eating breakfast (15 minutes)

Transportation from home to work (50 minutes)

So in this example we say that this person takes 2 hours and 5 minutes in the process since he wakes up and arrives to the office. If he intends to be in the office by 8 am he should wake up at 5:55am.

This app will help the user to get all of his goals “ON TIME” by scheduling a series of alarms or notifications of things that he should start doing or are pending to do in order to get his goal.

Intended User

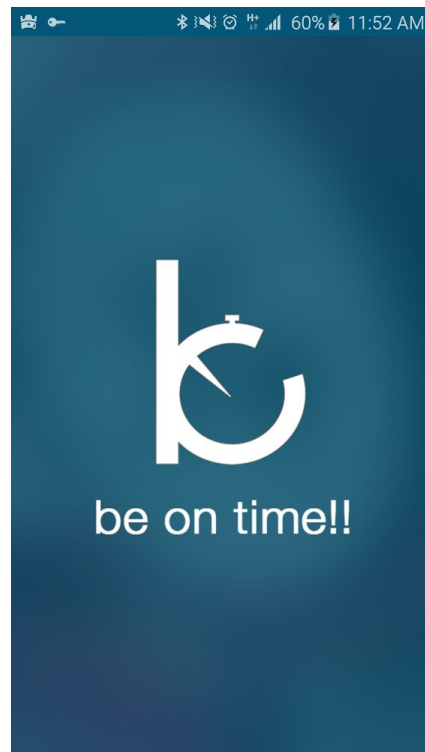
This app is meant for everyone that wants to get something done “ON TIME”, so anyone can take advantage of this app.

Features

- Schedule goals that want to be reached “ON TIME”
- Add tasks to this goals in order to adjust the time needed for the goal to be completed.
- Adjust the order of the task that needs to be done in order to complete the goal.
- Store this information in the device so the alarms can work off-line.
- Store information in a database (parse, **firebase**, etc) so we can have a backup of this info.
- Free and premium version of the app (free with advertising, premium with no advertising).
- Possibility to connect with google maps in tasks that require moving to a designated place.
- Possibility to enable/disable current goals without the need of deleting this items from the DB.
- AdMob advertising in free version.
- Rewards system (badges) in base of how many goals he has achieved “ON TIME”.
- Connectivity to social networks (facebook, google+, twitter) in order to share his achievements or to recommend the app.

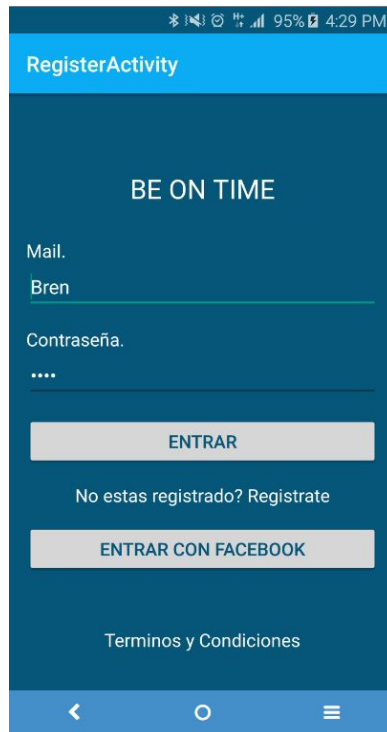
User Interface Mocks

Screen 1 (Splash screen)



This will be the initial screen each time the app is launched, just a splash screen.

Screen 2 (Login)

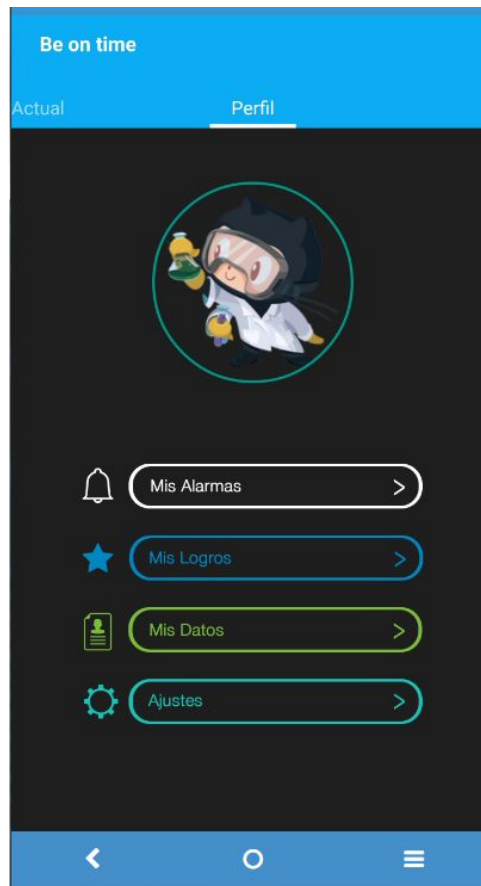


Login screen where you can enter your credentials and log into the app, once logged in the user and password will be stored (in a shared preference file) and auto-login will be enabled from preferences option.

This will have also the option to login with facebook account.

If no account has been created before it will let you register a new user.

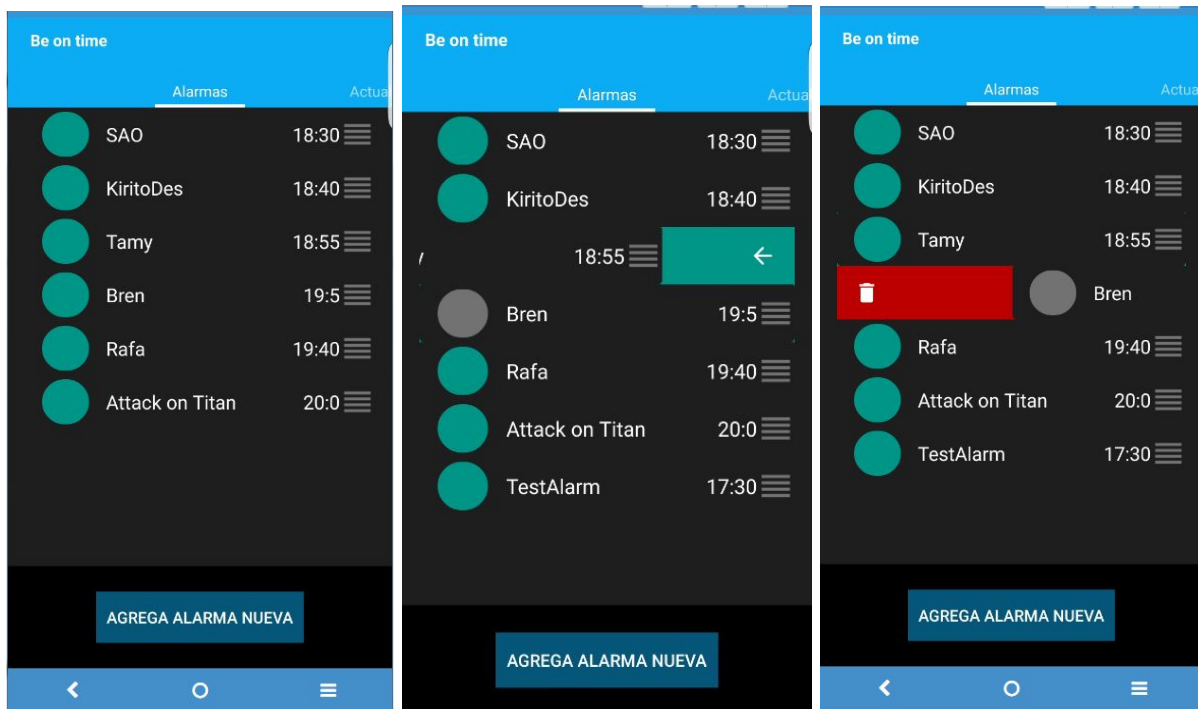
Screen 3 (Main Page - Profile page)



The main page will be a view pager with 3 tabs.

- Profile - This will show general info of the user, goals, achievements (badges), personal data and settings.
- Current - This will show the current running goal (if any).
- Goals - This will display all the goals the user has defined.

Screen 4 (Main Page - Goals page)



This screen will display all goals that the user has created.

This screen has some functionalities that are handle through swipe motions, a swipe to the left enables/disables the current goal, a swipe to the right deletes the selected goal and all his tasks, while dragging an item from the list in the toggle (button at the right side of each element) allow us to change the order in which the goals are displayed.

If you click in any goal it will display a new screen (Screen 5) where you can see and add new tasks to this goal.

Screen 5 (New Goal)

The image displays two side-by-side screenshots of a mobile application interface titled "Be on time".

Left Screenshot:

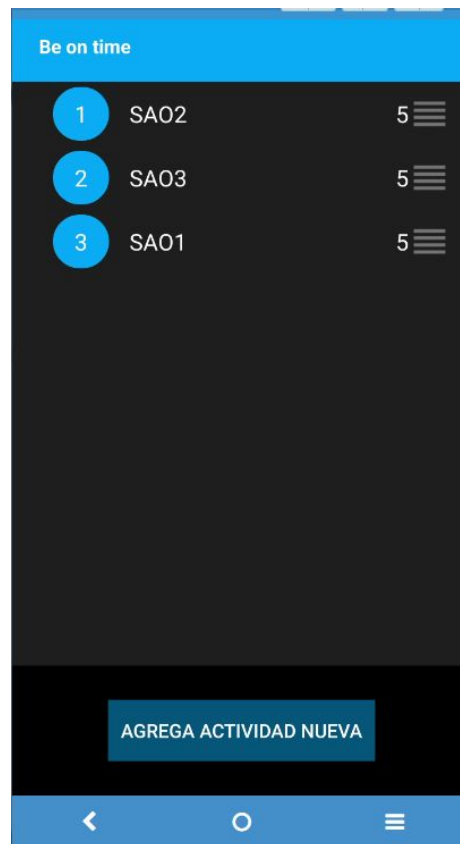
- Header: "Be on time"
- Text: "Nombre de la alarma:" followed by a text input field containing "Ej. Desayunar".
- Large digital clock display showing "4:46" with "AM" and "PM" indicators.
- A circular analog clock face with numbers 1 through 12. The number 4 is highlighted.
- Bottom section: "Frecuencia" (Frequency) dropdown menu showing "Una vez" (Once).
- Bottom navigation bar with back, home, and menu icons.

Right Screenshot:

- Header: "Be on time"
- A circular analog clock face with numbers 5, 6, and 7 highlighted.
- Configuration options:
 - "Frecuencia" (Frequency) dropdown menu showing "Una vez".
 - A slider for "Repetir cada:" (Repeat every) set to "10".
 - "Repetir cada:" (Repeat every) dropdown menu showing "0" and "Una vez".
 - A slider for "Repetir cada:" (Repeat every) set to "1".
 - "Días:" (Days) section with checkboxes for L, M, W, J, V, S, D.
 - "Termina:" (Ends) dropdown menu showing "Nunca" (Never).
 - "Resumen:" (Summary) section.
 - A "GUARDAR" (Save) button.
- Bottom navigation bar with back, home, and menu icons.

This screen is where you will define a new goal, you define a name, a target time, a frequency, an interval, the days you want it to work, if it has a final date and it will automatically fill a resume field where it'll display the preferences you just entered. After clicking on save button it will take you back to main screen (goals page) and display the new added goal.

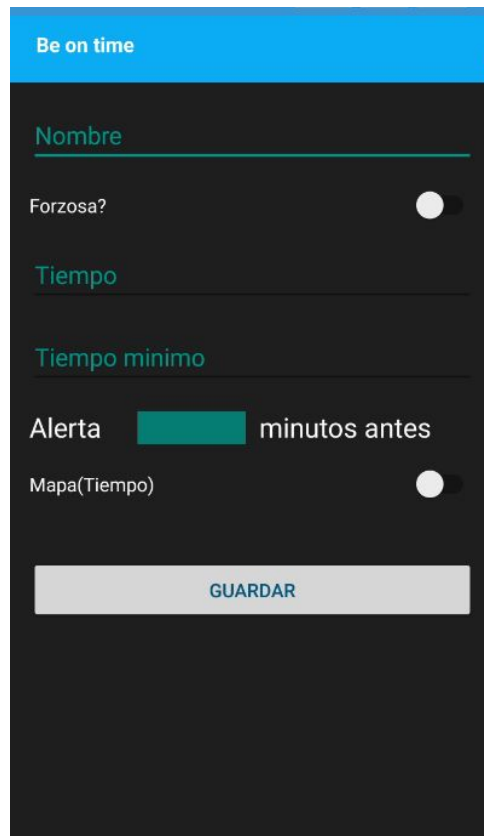
Screen 6 (Task List)



This screen will display all the tasks that needs to be executed in order to complete a goal, there is a toggle button at the right of each list item that allow us to drag an item and re order it.

There is also the functionality of swiping to the right to delete the selected task.

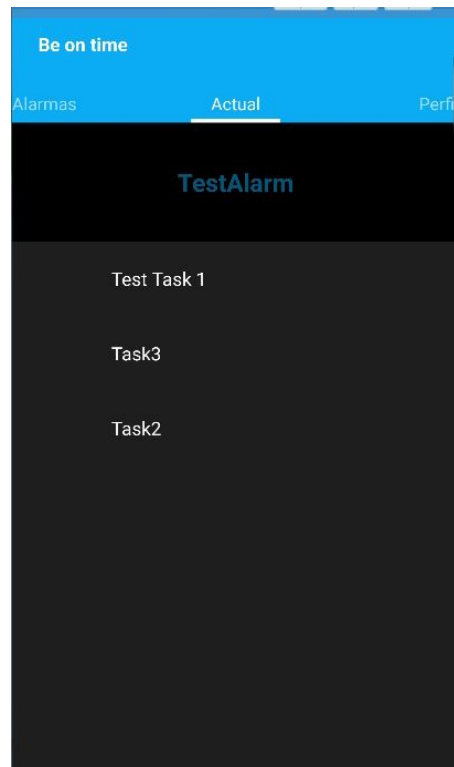
Screen 7 (New Task)



The screenshot shows a mobile application interface with a dark theme. At the top is a blue header bar with the text "Be on time". Below the header, the form contains the following elements: a teal-colored label "Nombre" followed by a text input field; a label "Forzosa?" followed by a toggle switch that is currently turned off; a teal-colored label "Tiempo" followed by a text input field; a teal-colored label "Tiempo minimo" followed by a text input field; a label "Alerta" followed by a teal-colored numeric input field containing the number "5" and the text "minutos antes"; a label "Mapa(Tiempo)" followed by a toggle switch that is currently turned off; and a large, light gray button at the bottom with the text "GUARDAR" in blue capital letters.

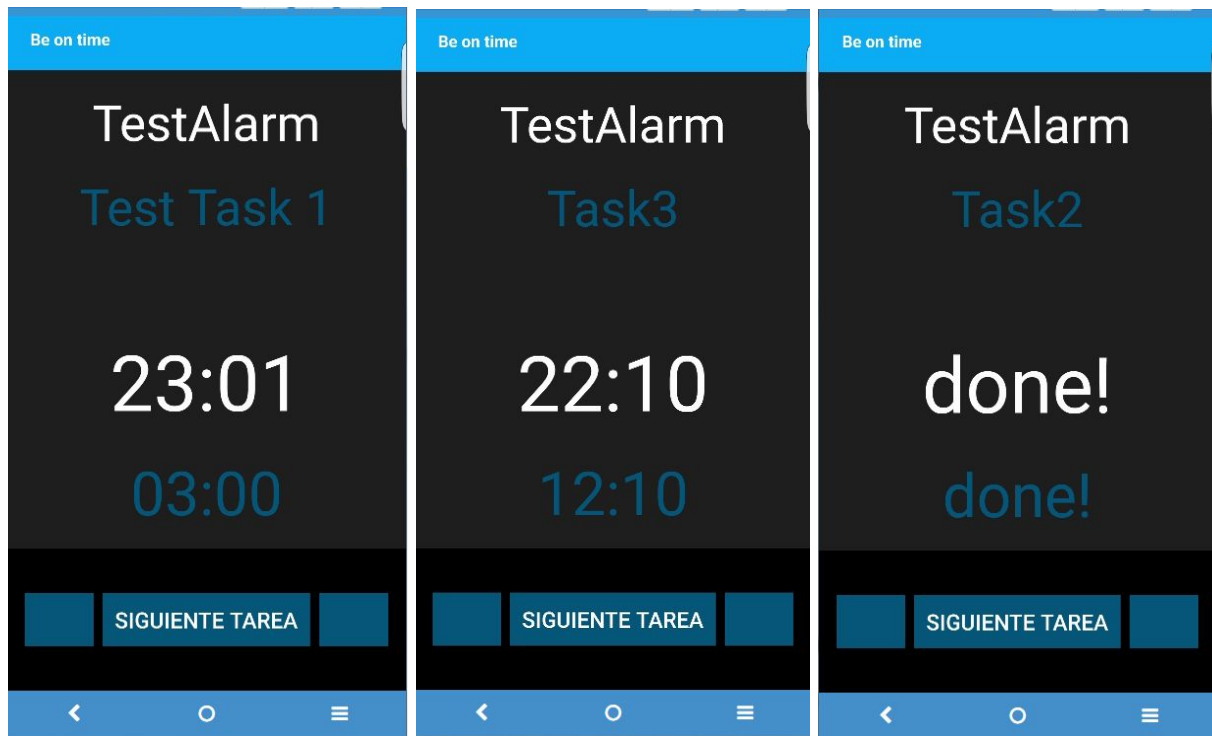
This is the screen where you create new tasks, you need to define a name, if the task is mandatory or not, the time that you normally use to complete this task, the minimum amount of time in which it can be completed.

Screen 8 (Main page - Current Alarm)



This screen is the center tab from the main page (navigation drawer) if there is an active alarm running when you enter the app. You will be redirected to this page. This will display the current alarm, and the tasks that form part of it, highlighting the task that you should be doing in this moment.

Screen 9 (Active Alarm)



This screen will display the info of the active goal/alarm, the time remaining for the single task and for the general goal. Once you finish your task you can press next task button and it will display the next task to be done and update the time you have to do that.

After all tasks have been finished it should display done! And store the info of your time. This is just for the badges system that will be implemented.

Key Considerations

How will your app handle data persistence?

The data persistence will be handled with a content provider that I will made by myself, also there will be a service to store the alarm info into a cloud db (Mongo DB, parse or firebase are my options now).

Describe any corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

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

- Glide. I will use glide for handling image loading, this is nos because it is better but in my personal opinion has some functionalities that piccaso doesn't have, like animated gif loading into an imageview.
- Butterknife. I really like this library because the binding of views or resources a lot of times makes the code more readable and easy to understand.
- AppCompat. To support older versions and still being able to apply some material design features.
- Parse (not decided yet). To handle cloud db storage.
- Facebook. To allow user to signin with their facebook accounts.

Next Steps: Required Tasks

Task 1: Project Setup

- Add all the previous defined libraries in order to being able to start coding the app.
- Create Free and Paid gradle flavors.

Task 2: Create splash screen

- Create a new project (BeOnTime)
- Create a new Activity(Splash activity)
- Implement logic and functionality for this splash activity.
- Create LoginActivity
- Handle navigation to show LoginActivityafter some seconds of having showed the splash activity.

Task 3: LoginPage

- Create Layout for login page.
- Add facebook sdk in order to allow login through facebook account.
- Implement cloud DB storage so you can add new users and have a record of the existing users.
- Implement register login.
- Implement login logic and functionality.
- Connect login response with Main Page.
- Handle error messages in login.

Task 4: Content Provider and Data Storage

- Create content provider for internal data storage.
- Implement the needed code to insert and retrieve information.
- Define the DB Schema.
- Replicate same schema in Parse or selected db storage.

Task 5: Main page (View Pager)

- Create a layout with a viewpager that will act as the main page.
- Design the layouts of each tab that will be implemented for the viewpager.
- Create code to animate transitions between pages.

Task 6: Profile page

- Create layout for the profile screen.

- Functionality to load a profile image.
- Design of preferences page
- Create rewards page with description of what needs to be done in order to get that badge.
- Create layout to capture personal data for the user, we can also edit this info if it's needed.

Task 7: Goal Page

- Create a layout to be used in the viewpager that will act as the Goals page.
- Implement recyclerview to contain the list that we'll display to the user.
- Implement Dragging and swiping functionality in the recyclerview.
- Create the TaskListActivity that will contain all tasks that are generated for a specific goal.
- Navigation to TaskListActivity or to NewGoalActivity.
- Handle error messages.

Task 8: Current Alarm Page

- Create a layout to be used in viewpager for displaying current active alarm.
- Recyclerview that will display all the task that the active alarm has.
- Highlight functionality to indicate which is the active task in the goal.
- Navigation to CurrentAlarmDetail activity.

Task 9: New Goal Activity

- Create a layout for the new goal activity.
- Use android widgets to better UX.
- Implement logic for creating new goal.
- Store new goal (locally and remote)
- Implement AdMob banner in free flavour.

Task 10: Task Details Activity

- Create layout for displaying a list of task that are in a goal.
- Implement functionality to drag and drop and change order of new tasks.
- Implement logic to navigate to NewTaskActivity.
- Implement logic to swipe (delete) functionality.

Task 11: New Task Activity

- Create layout for NewTaskActivity that will allow us to create new tasks.
- Implement logic to handle all data entered by the user.
- Store the new task and navigate back to TaskLists page when saving one task.
- Implement AdMob banner in free flavour.

Task 12: Current Alarm Detail

- Create layout for CurrentAlarmDetail that will display the info of a current active alarm.
- Implement 2 countdown timers (one for the general goal and one for the task's) that will display the remaining time left to complete the task/goal.
- Implement logic to cancel pendingIntents in case a task is finished before time, also update the data so the UI can adapt and display the correct status.

Task 13: Implement Google Play Games

- Implement Google Play Games services to add rewards mechanism to the app, this to gamifying the user experience and trying to get him involved with accomplish his goals on time.