Simple Habits

Version 1.4

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 31/10/16 | 1.0 | Initial Creation | Benedikt Bosshammer |
| 02/11/16 | 1.1 | Mockups | Benedikt Bosshammer |
| 13/11/16 | 1.2 | CRUD Basic Activity Diagram | Benedikt Bosshammer |
| 15/11/16 | 1.3 | Create and Read Activity Diagram | Benedikt Bosshammer |
| 16/05/2017 | 1.4 | Added Function Points | René Penkert |

Table of Contents

1. Select Calendar 4

1.1 Brief Description 4

2. Flow of Events 4

2.1 Basic Flow 4

3. Special Requirements 7

4. Preconditions 7

5. Postconditions 7

6. Extension Points 7

7. Function Points 7

)

# Select Calendar

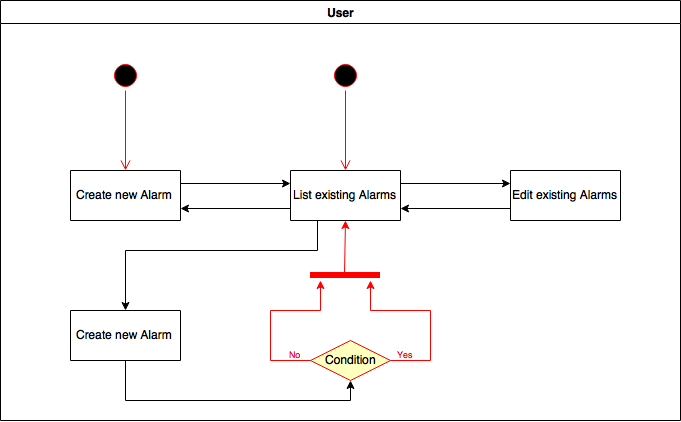
## Brief Description

This Use Case provides the possibility for the users to select the OS calendars containing the appointments for wake-up time calculation.

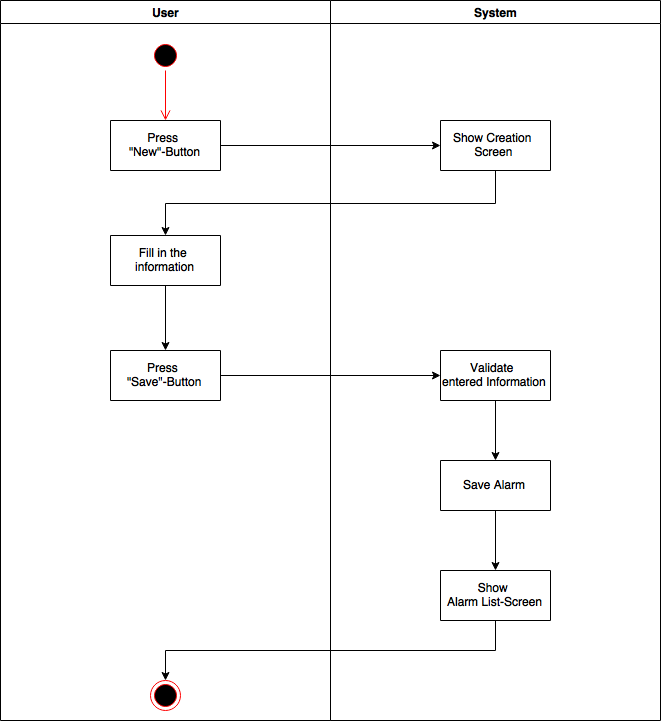
# Flow of Events

## Basic Flow

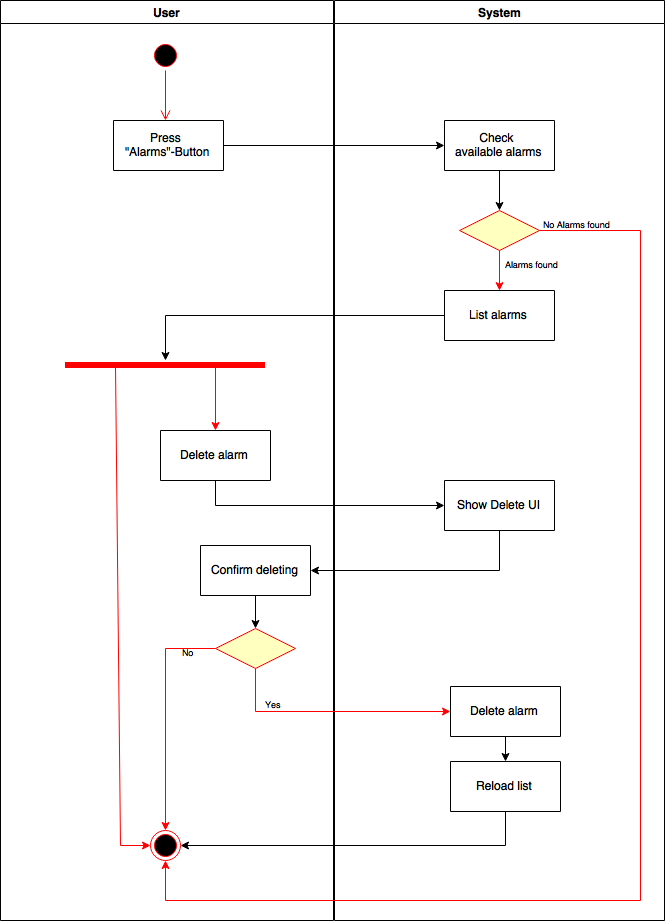
The process of managing the alarms can be represented by a CRUD activity diagram. The User can update either create a new alarm (Create), display all alarms created before (Read), delete on of these alarms (Delete) or edit an alarm (Update)



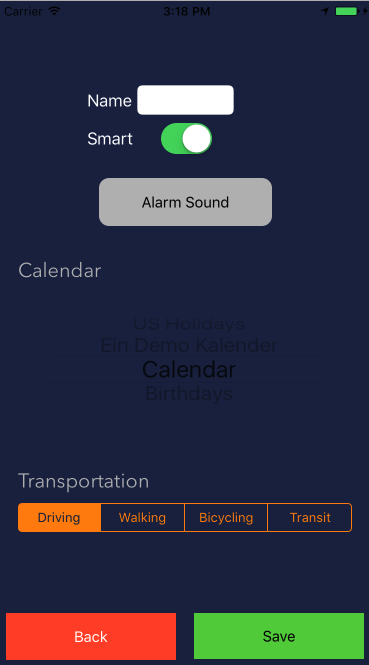
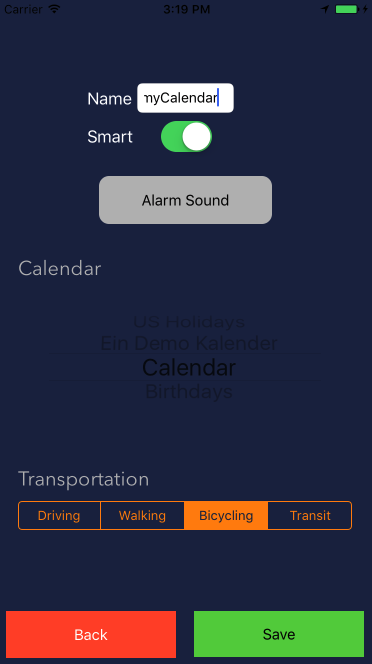
A closer look at the activities in case the user decides to create a new alarm:



The following activity diagram shows the event in case the user decides to read an existing alarm.



iOS Mockups:

# Special Requirements

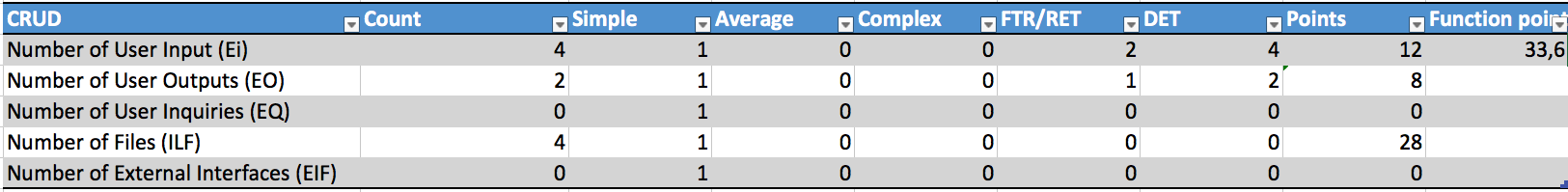
# Preconditions

To add a Smart Alarm, the User needs to give Access to the Calendar so that the SmartAlarm have access to the appointments in the specified Calendar.

# Postconditions

# Extension Points

# Function Points



# Testing

[UI-Test](https://github.com/renpen/SimpleHabits/blob/master/xCodeProject/AlarmClockUITests/CreateAlarm.swift)

[Unit-Test](https://github.com/renpen/SimpleHabits/blob/master/xCodeProject/AlarmClockTests/AlarmCoreData.swift)