
penbo Simplicity

Software Requirements Specification

Version 2.0

Smart Clock	Version: 2.0
Software Requirements Specification	Date: 22/10/16

Revision History

Date	Version	Description	Author
22/10/16	1.0	Initial creation	Benedikt Bosshammer
25/11/16	2.0	Reworked UCD	Benedikt Bosshammer

Smart Clock	Version: 2.0
Software Requirements Specification	Date: 22/10/16

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms, and Abbreviations	4
1.4	References	4
1.5	Overview	4
2.	Overall Description	5
3.	Specific Requirements	5
3.1	Functionality	5
3.1.1	Only use specified calendars	6
3.1.2	For time calculation use location from calendar appointment	6
3.1.3	Route calculation via Google Maps API	6
3.1.4	Spotify Integration for alarm sound	6
3.2	Usability	6
3.2.1	Background Service	6
3.2.2	iOS Usability Standards	6
3.2.3	As less as possible manual user input	6
3.3	Reliability	6
3.3.1	Only local runtime	6
3.3.2	Background Job Reliability	6
3.4	Performance	6
3.4.1	Catch weak network access	6
3.5	Supportability	6
3.6	Design Constraints	6
3.7	On-line User Documentation and Help System Requirements	7
3.7.1	Guided introduction tour	7
3.8	Purchased Components	7
3.8.1	Apple Developer License	7
3.9	Interfaces	7
3.9.1	User Interfaces	7
3.9.2	Hardware Interfaces	7
3.9.3	Software Interfaces	7
3.9.4	Communications Interfaces	7
3.10	Licensing Requirements	7
3.11	Legal, Copyright, and Other Notices	7
3.12	Applicable Standards	7
3.12.1	iOS App Standards	7
4.	Supporting Information	7

Smart Clock	Version: 2.0
Software Requirements Specification	Date: 22/10/16

Software Requirements Specification

1. Introduction

1.1 Purpose

This SRS will give an overview for the requirements of this project.

1.2 Scope

This document is for internal use only and shall give directives for the development of this project.

1.3 Definitions, Acronyms, and Abbreviations

SRS	Software Requirements Specification
tbd	to be determined
n/a	not applicable

1.4 References

UC Document Select OS Calendar:

<https://github.com/renpen/SimpleHabits/blob/master/SoftwareEngineering/Published/UseCaseSelectCalendars.pdf>

UC Document Manage Alarms (CRUD):

<https://github.com/renpen/SimpleHabits/blob/master/SoftwareEngineering/Published/UseCaseManageAlarms.pdf>

Software Architecture Document:

<https://github.com/renpen/SimpleHabits/blob/master/SoftwareEngineering/Published/Software%20Architecture%20Document.pdf>

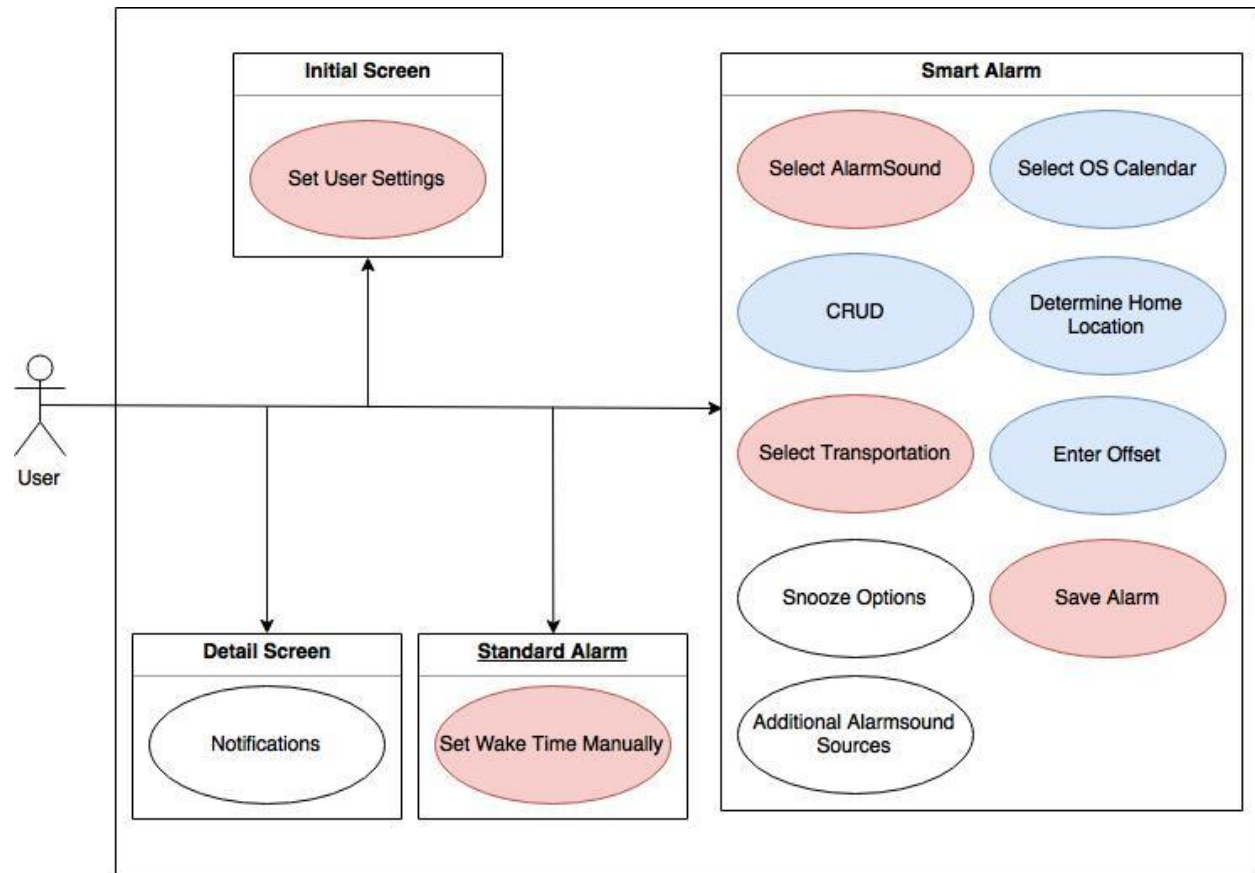
1.5 Overview

The following chapters are about our vision and perspective, the requirements for deploying the software on our side and accessing the software on the user side, the demands we have, licensing and the technical realization of this project.

Smart Clock	Version: 2.0
Software Requirements Specification	Date: 22/10/16

2. Overall Description

See 'The Vision' on our website (<http://penbo-simplicity.de/wordpress/index.php/vision>)



3. Specific Requirements

3.1 Functionality

Set alarm according to calendar appointments

Smart Clock	Version: 2.0
Software Requirements Specification	Date: 22/10/16

3.1.1 *Only use specified calendars*

3.1.2 *For time calculation use location from calendar appointment*

If there is no location entered, ask for user

3.1.3 *Route calculation via Google Maps API*

3.1.4 *Spotify Integration for alarm sound*

3.2 Usability

Ideally the app is simple enough to be understood right away but we plan on adding a short guided tour through it that explains the usage.

3.2.1 *Background Service*

Sensitize the user for known weaknesses like the need to run the app in the background and to restart it after device reboot.

3.2.2 *iOS Usability Standards*

3.2.3 *As less as possible manual user input*

3.3 Reliability

3.3.1 *Only local runtime*

Since this is an app that runs exclusively local server uptime and similar things are not an issue. Instead reliability is completely dependent on a good update chain.

3.3.2 *Background Job Reliability*

The app should run reliable due to the importance of setting the alarm right

3.4 Performance

3.4.1 *Catch weak network access*

tbd

3.5 Supportability

3.5.1.1 *Appstore*

To automatically push production updates to users the application will be deployed in the AppStore when finished. This makes updates easier and is the known standard for iOS applications.

3.5.1.2 *Coding standards*

The standard coding conventions for Swift will be used. This means for example not to use ; to end a statement.

3.5.1.3 *Variable Names*

To make our coding easier to ready for maintenance we want to use code conventions. Due to the fact that variables in swift are declared with a type but with the universal statement 'var' we will put the type at the beginning to have this type identification allows us to see the type within the variable name even in deeper coding.

3.5.1.4 *Lines per function*

3.6 Design Constraints

The app will only use standard UI-Elements from the Xcode Library for iOS.

Smart Clock	Version: 2.0
Software Requirements Specification	Date: 22/10/16

3.7 On-line User Documentation and Help System Requirements

3.7.1 Guided introduction tour

Show every needed information and customizable settings.

3.8 Purchased Components

3.8.1 Apple Developer License

Only needed for AppStore release

3.9 Interfaces

3.9.1 User Interfaces

Touchscreen on iPhones and iPads

3.9.2 Hardware Interfaces

n/a

3.9.3 Software Interfaces

3.9.3.1 Google API

3.9.4 Communications Interfaces

tbd

3.10 Licensing Requirements

3.11 Legal, Copyright, and Other Notices

No notices yet. Maybe necessary once we use better graphics and want to put it in the AppStore.

3.12 Applicable Standards

3.12.1 iOS App Standards

4. Supporting Information

tbd