POLITECNICO
MILANO 1863

Release date 11-07-2017

Design and Implementation of Mobile Applications: EasyMorning

Design document

V2.0

Dimitar Anastasovski 10515463
Petar Korda 10518640

Contents

Chapter 1

Introduction

1.1 Purpose

This document is intended to help understand and evaluate design and prototyping steps taken for the mobile application “EasyMorning”, explaining both the application domain and the system that you want to accomplish. It explains the functional features of the “EasyMorning”, along with interface details, design constraints and related considerations such as performance characteristics. This project is implemented as an outcome of the project “Design and implementation of mobile applications”.

1.1.1 Intended audience

This document is intended for all individuals participating in and/or supervising the project:

Expected audience of this document is the developers and anyone who in-tends to develop on this program

Developers who can review project's capabilities and more easily understand where their eforts should be targeted to improve or add more features to it (it sets the guidelines for future development).

Project testers can use this document as a base for their testing strategy as some bugs are easier to find using this document. This way testing becomes more methodically organized.

End users of this application who wish to read about what this project can do.

1.2 Scope

The main accent is to simplify and optimize alarm application that we use every day with additional features that will make the application more usable, user friendly and customizable. This application is developed for IOS devices and it’s only supported for this kind of devices. This application can be used by anyone who previously will download and install this application on their mobile phone. After the installation welcome screen will appear with the logo of the application and shortly after that will appear the navigation. From here, users can schedule their alarms and select some of the features that are given to the users. They can choose if they want to have the latest news that are feed from the most visiting news site, weather forecast for the current city or city that they want to choose and they can enable playing music that is feed directly from Spotify that will play after the alarm triggers and the user is “awake”.

1.3 Goals

The system has to provide this features:
[G1] Only users that poses IOS phones can use the service

[G2] Users can use this application if and only if they previously installed this application from the AppStore

[G3] Users can schedule alarms
[G4] Users can read the news
[G5] Users can see the weather forecast
[G6] Users can play Spotify playlists
[G7] Users are able to combine this
features

1.4 Document overview

This document is divided in four parts with clean and non-ambiguity description of the whole system.

Chapter 1: General description and basic informartion of the system

Chapter 2: Explanation of the main functionalities with constraints, assumptions and hardware dependencies.

Chapter 3: Specifications of the system functionalities and non-system func-tionalities possible scenarios

Chapter 4: Use cases and UML models

Chapter 2

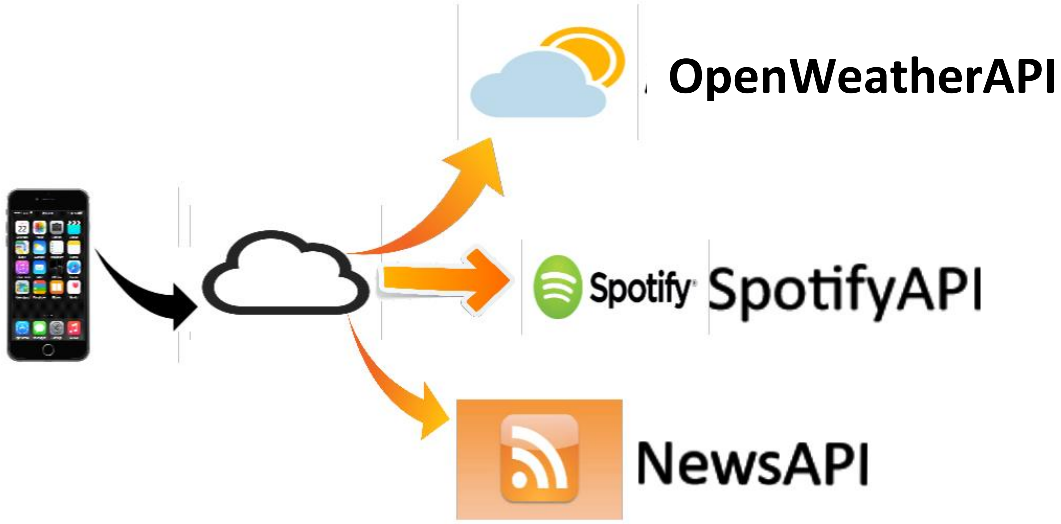
Overall description

2.1 Architecture design

For this project bottom-up approach was used where carefully were made decisions for strategies and selection of the priorities to be pursued in this project. Also separating in several sub- systems makes the whole system more robust, scalable and easier for understanding the issues in implementing functionalities and to separate, logically, groups of functionalities and state clearer their interaction.

System is separated in four sub-systems:

- Alarm models, controllers and views
- Spotify API
- Weather API (OpenWeatherAPI)
- News reader API(NewsAPI)



2.2 Use case functional requirements analysis

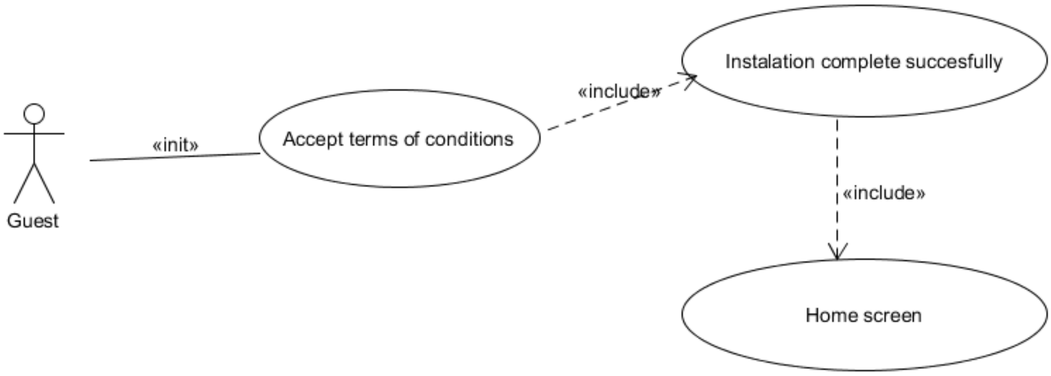
This chapter captures the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. Also we have goal-oriented set of interactions between external actors and the system under consideration. Actors are parties outside the system that interact with the system. Thus, we provide who (actor) does what (interaction) with the system, for what purpose (goal), without dealing with system internals.

We can identify one group of actors involved in this application.

- Users –who had previously downloaded the application

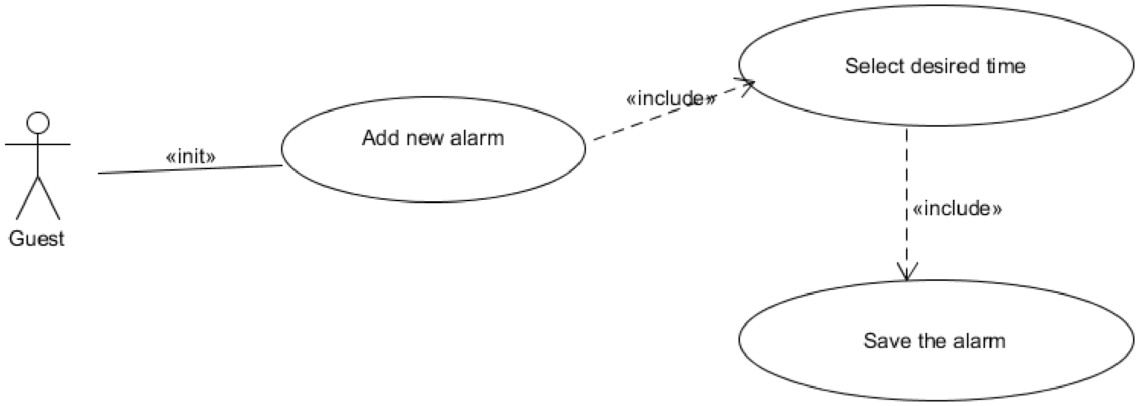
2.2-1 Downloading and installing the application

Actor	Users
Goal	Download and install the application
Event flow	1.Users navigates to the app store icon 2.Users search for the app 3.Users accept tearms of conditions 4.Users installs the application
Output conditions	Users succesfully ends the procedure of installing the application and can use it for now on
Exception	1.Users already had the application 2.Other hardware dependencies



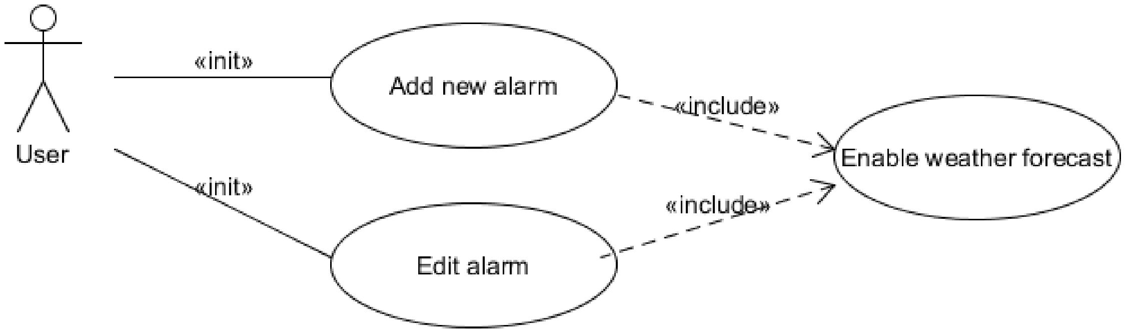
2.2-2 Scheduling alarm

Actor	User
Goal	Can schedule alarm on specific time
Event flow	1.User clicks on the “+” icon for making new alarm 2.User select the desired time 3.User clicks to “Save” button
Output conditions	Alarm is schedule
Exception	1.User can’t select past time 2.User can just schedule alarm for the next day



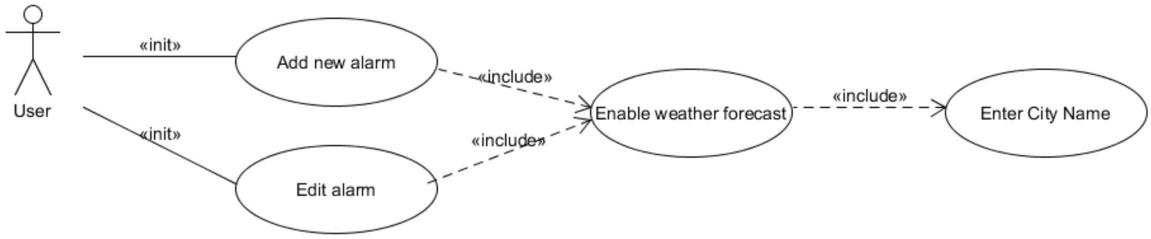
2.2-3 Enable weather forecast

Actor	User
Goal	Enabling weather forecast
Event flow	1.User clicks on the “+” icon for making new alarm 2.User enables Weather Forecast 3.User clicks to “Save” button
Output conditions	Weather Forecast is enabled
Exception	None



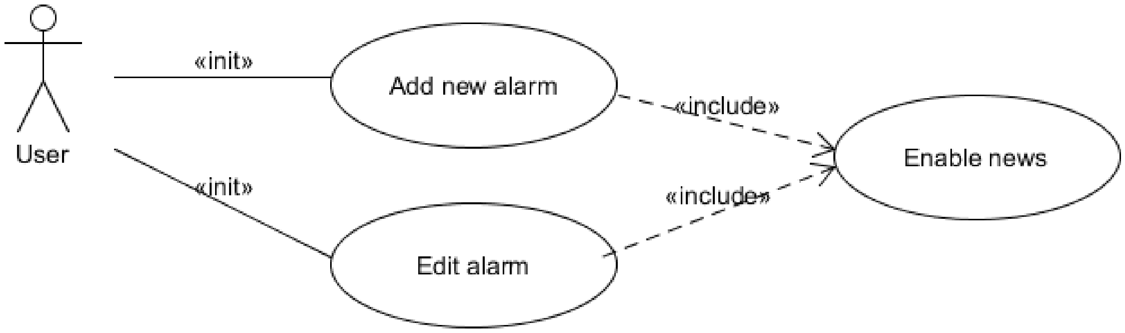
2.2-4 Enter city for weather forecast

Actor	User
Goal	Entering specific city for weather forecast
Event flow	1.User clicks on the “+” icon for making new alarm 2.User enables Weather Forecast 3.User enters city name 3.User clicks to “Save” button
Input conditions	1.User previously enabled Weather Forecast
Output conditions	Entered city weather forecast
Exception	1. Weather forecast previously enabled 2.City doesn’t exist



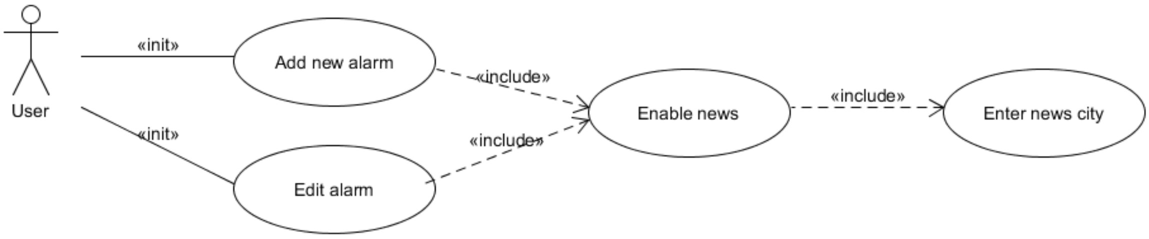
2.2-5 Enable news

Actor	User
Goal	Enabling latest news
Event flow	1.User clicks on the “+” icon for making new alarm 2.User enables News 3.User clicks to “Save” button
Output conditions	News feature is enabled
Exception	None



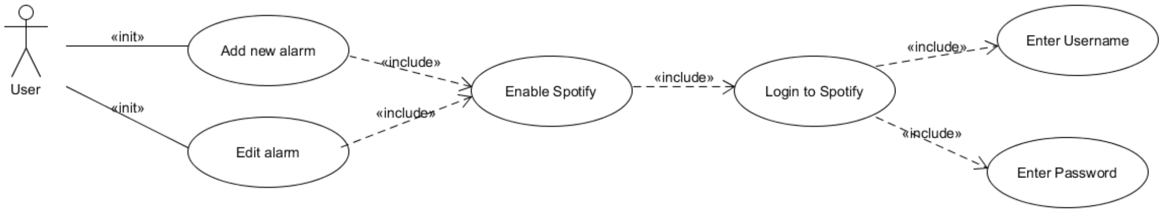
2.2-6 Enter news site

Actor	User
Goal	Entering specific news site
Event flow	1.User clicks on the “+” icon for making new alarm 2.User enables News 3.User enters site name 4.User clicks to “Save” button
Input conditions	1.User previously enabled News
Output conditions	News from entered site will be shown
Exception	1.Entered site doesn’t exist



2.2-6 Enable Spotify

Actor	User
Goal	Enable Spotify
Event flow	1.User clicks on the “+” icon for making new alarm 2.User enables Spotify 3.User clicks to “Save” button
Input conditions	1.User have Spotify account and is log in
Output conditions	Spotify is enabled
Exception	1.Entered username and password are incorrect



2.2-7 Permissions for notifications

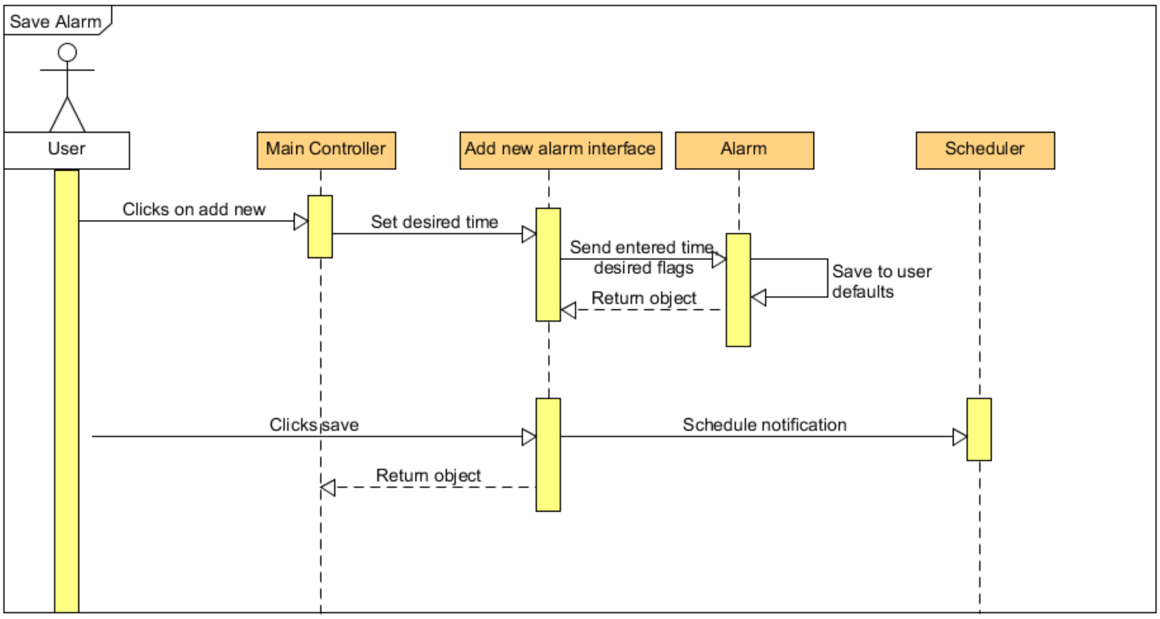
Actor	User
Goal	Allowing permissions for showing notifications
Event flow	1.User clicks on “Agree” button
Input conditions	None
Output conditions	Application will show notifications
Exception	1.User clicked on “Don’t agree”

2.3 Sequence diagrams

In this chapter we will model the flow of logic within our system in a visual manner that will enable us to document and validate our logic for further analysis and design purpose. We will present the most important interactions between the system and the user.

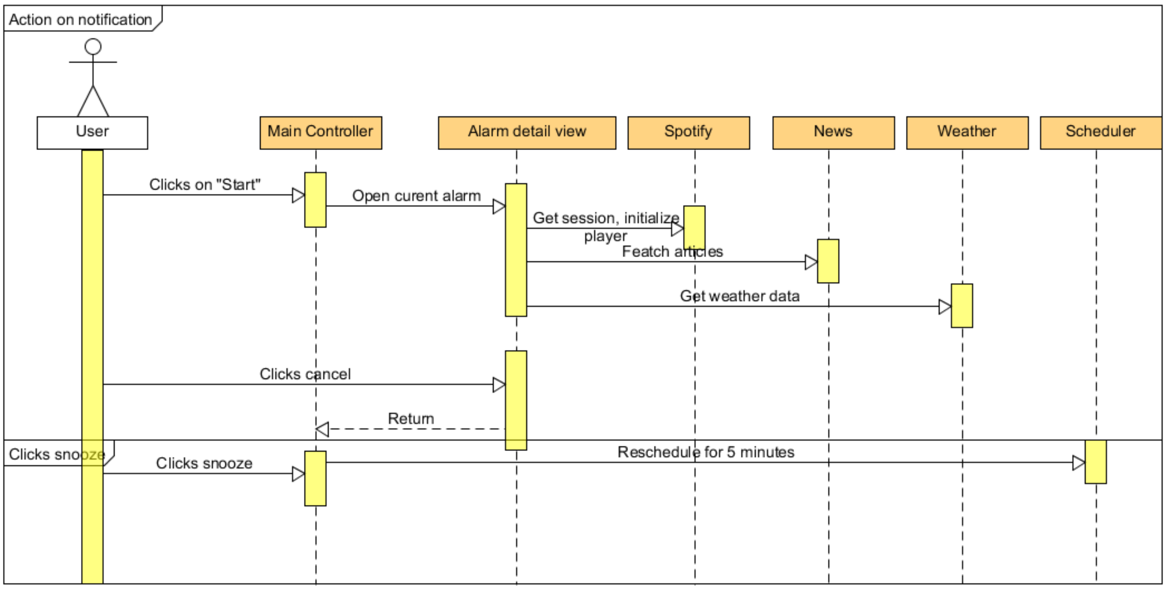
2.4 Set and save alarm

When user first land on the initial screen he/she will need to add alarm. User click on the "+" button that will open new screen when other options will be present. After the screen is loaded the user can select the desired time when he/she wants to trigger the alarm. The user can select the time in the format "hh:mm". The "Alarm" instance saves the entered time in user defaults and when the user clicks on the "Save" button the "Scheduler" is scheduling notification based on the entered time. Also the user can select some of the additional features that the app can show. He can select news, weather forecast and music streaming from Spotify. For the weather forecast he can also choose different city that he wants to get weather information by entering the city in the text box, if he don't do that current location will be shown.



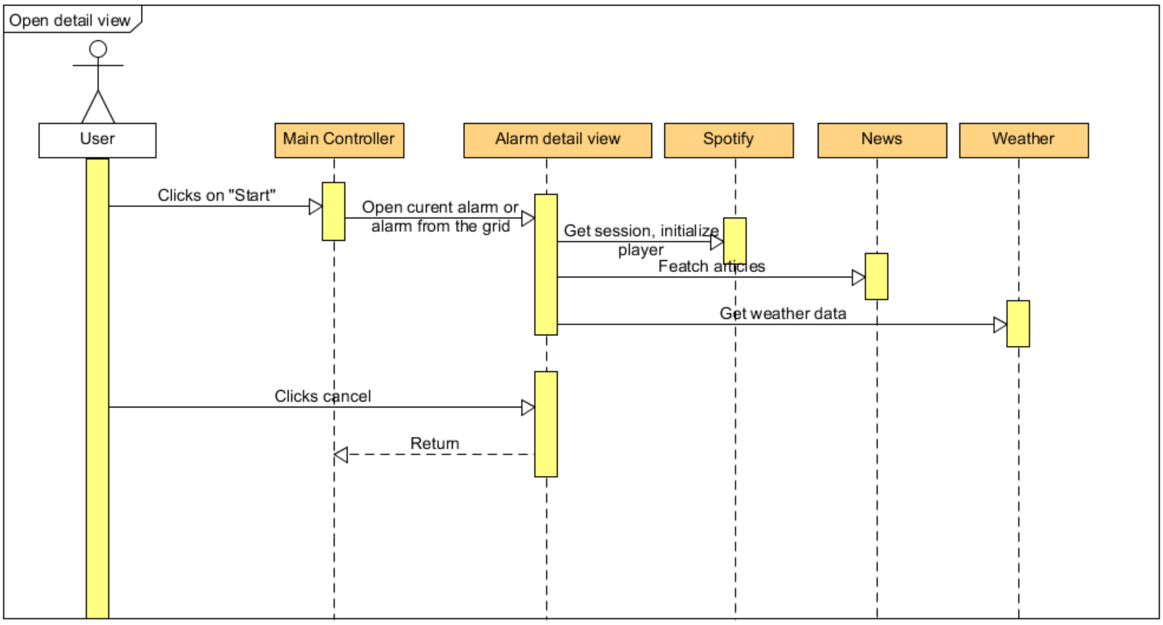
2.5 Notification actions

When the notification is fired the user can chose between two options, “Start” and “Snooze”. If he clicks on “Snooze” button it will reschedule the alarm for 5 minutes. If he clicks on “Start” button the Main Controller will open the current alarm and will show the detail alarm view based on our previously selected preferences.



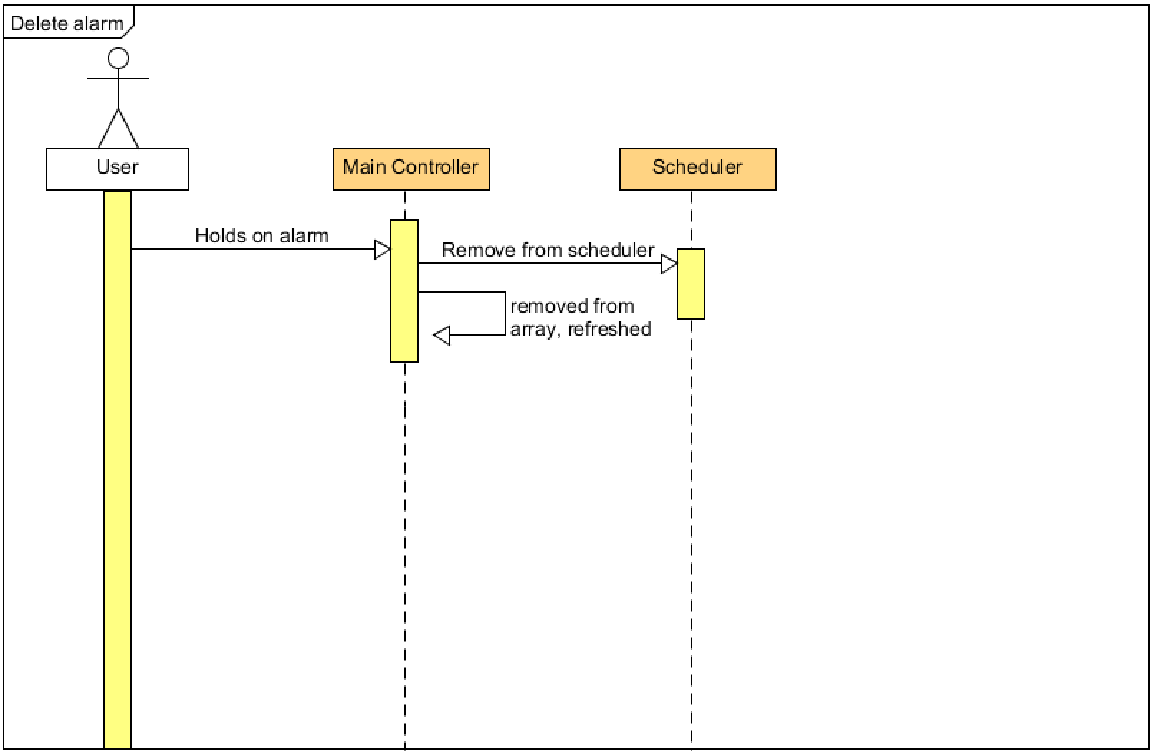
2.6 Open detail view

User can open the detail view when the notification is fired by clicking on the “Start button” or he can open it when he is in the grid view. When he opens the detail view he can see the previously entered preferences shown in three different tabs positioned at the bottom of the screen.



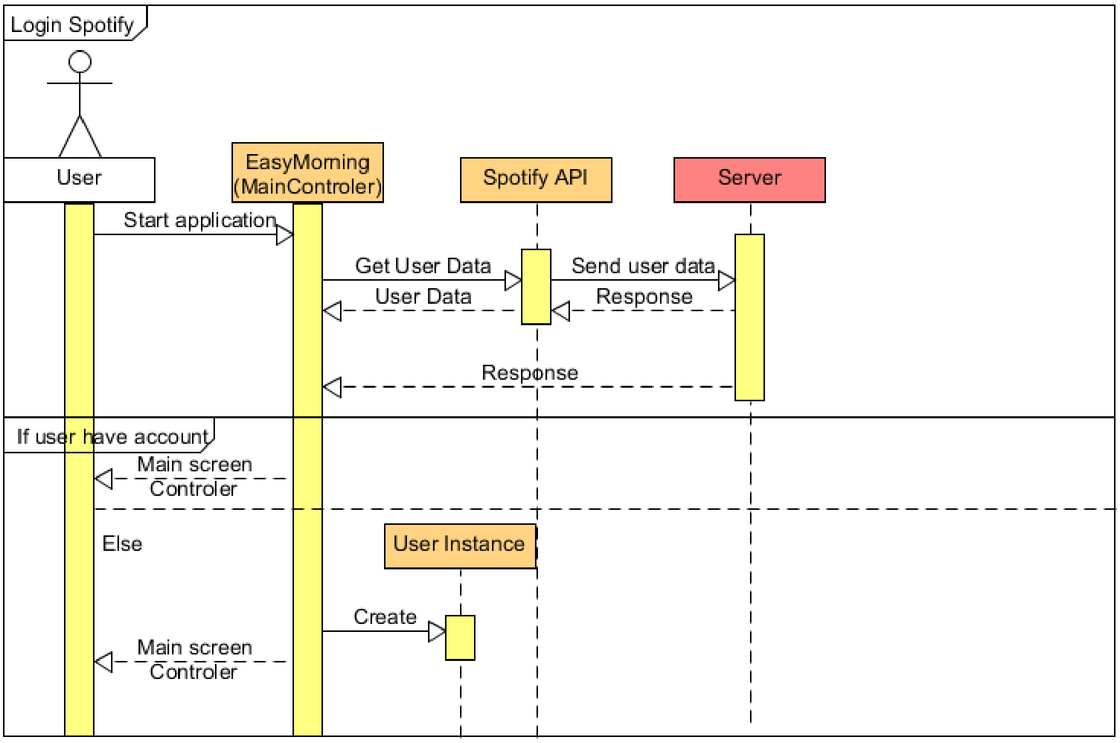
2.7 Delete alarm

User can delete alarms from the grid view by holding on the alarm. After two seconds a pop up notification will be shown that will ask the user if he is sure that we wants to delete the alarm. If he choose yes the alarm will be deleted from the "Scheduler".



2.8 Spotify login

To be enabled the music option for the user he needs to have an account on “Spotify” or he will be redirect to create a new one.

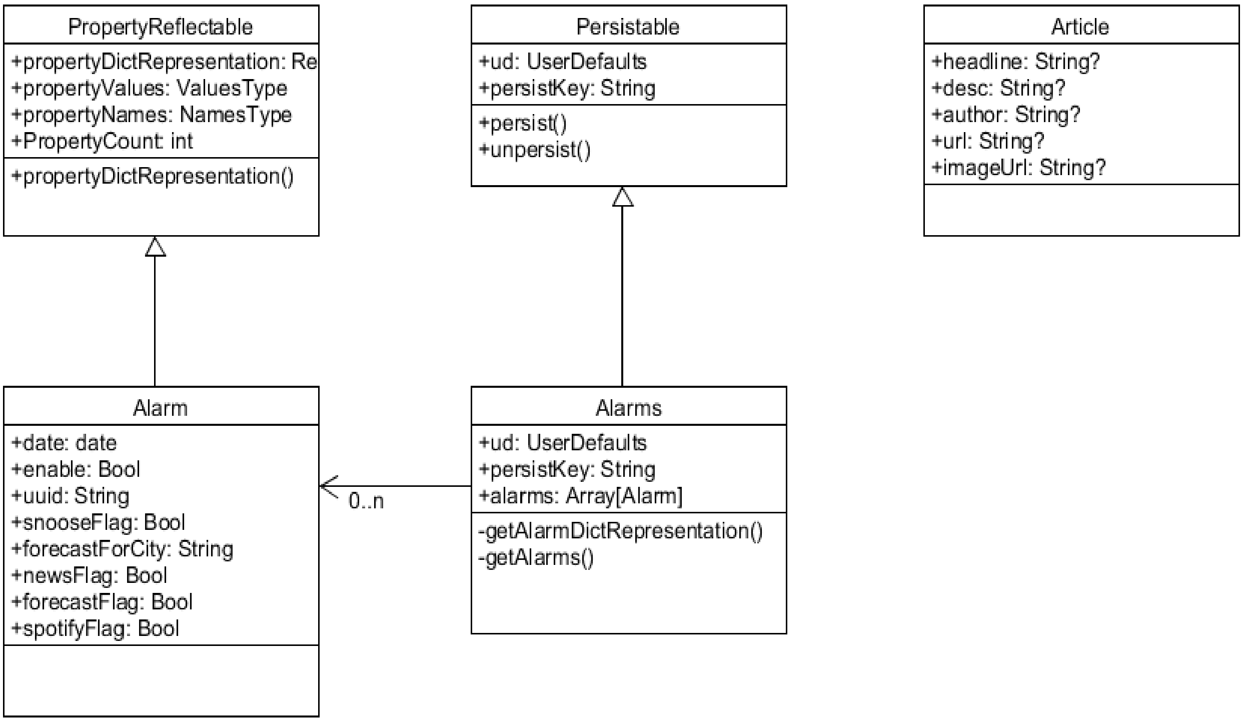


3 Class diagram

In this part we will present the object oriented analysis and design of our application. Class diagrams will show the classes of the system, their interrelationships (including inheritance, aggregation and association) the operations and the attributes of the classes.

3.1 Models

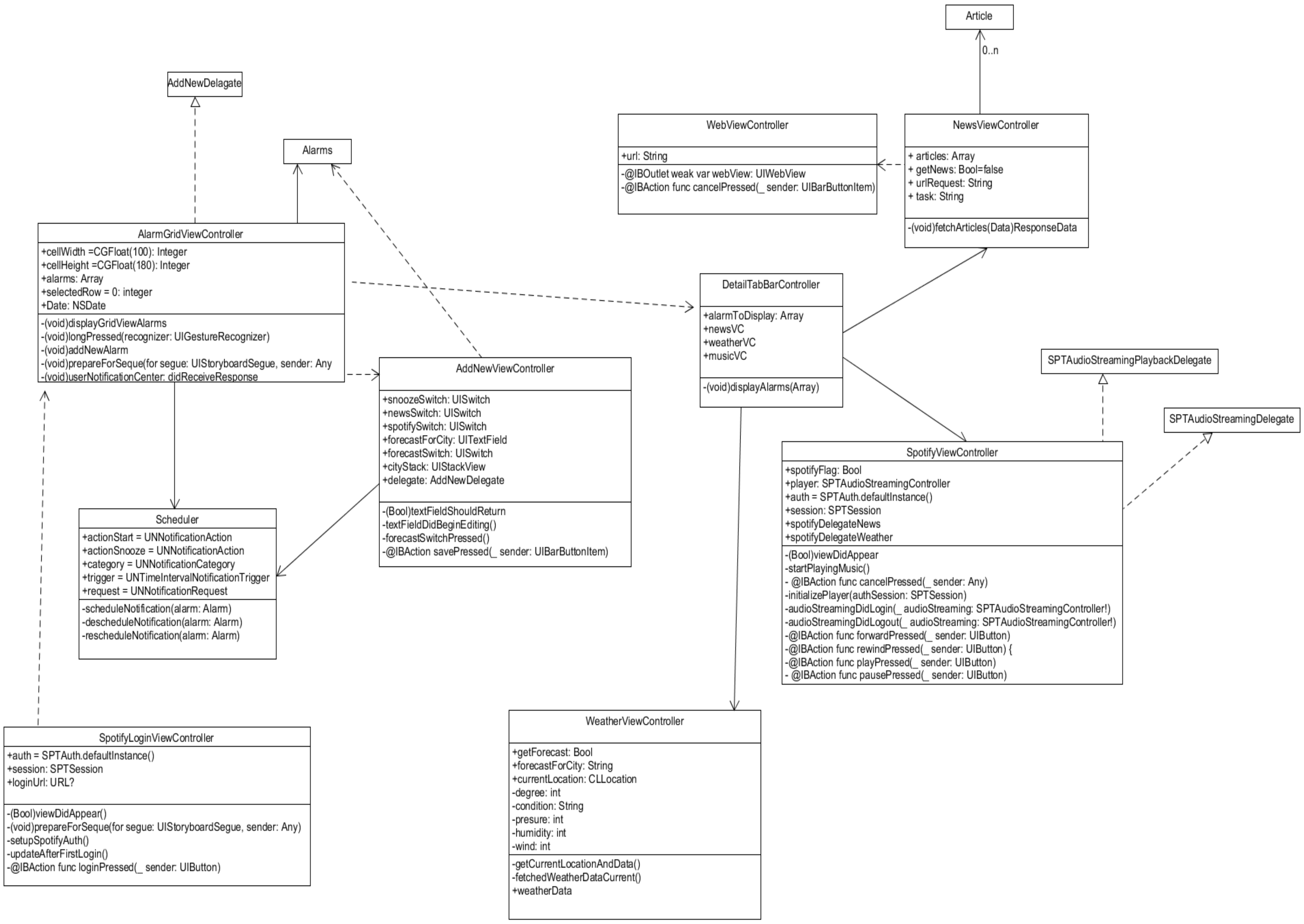
In this picture below Persistable and PropertyReflectable are protocols. Persistable is used for storing data in user defaults. Alarms is class implementing handling of alarms and includes an array of alarms, handling them and storing in user defaults.



3.2 Controlers

In the following text we will talk briefly about some of the controllers in the application:

- SpotifyLoginViewController- controls the login screen and the user logs in he is redirected to the main view
  .
- AlarmGridViewController- this is the main controller responsible for displaying the array of user alarms. Implements AddNewAlarmProtocol (delegate for AddNewVIewController)
- DetailTabBarController- contains three controllers corresponding to three different screens (Spotify,news,weather forecast)
- SpotifyViewController- implements protocols from the Spotify SDK and controls playback options
- NewsViewController- controls the table view of the fetched news articles
- WeatherViewController- fetches the weather data and displays them on the forecast screen
- AddNewViewController- is responsible for adding a new alarm

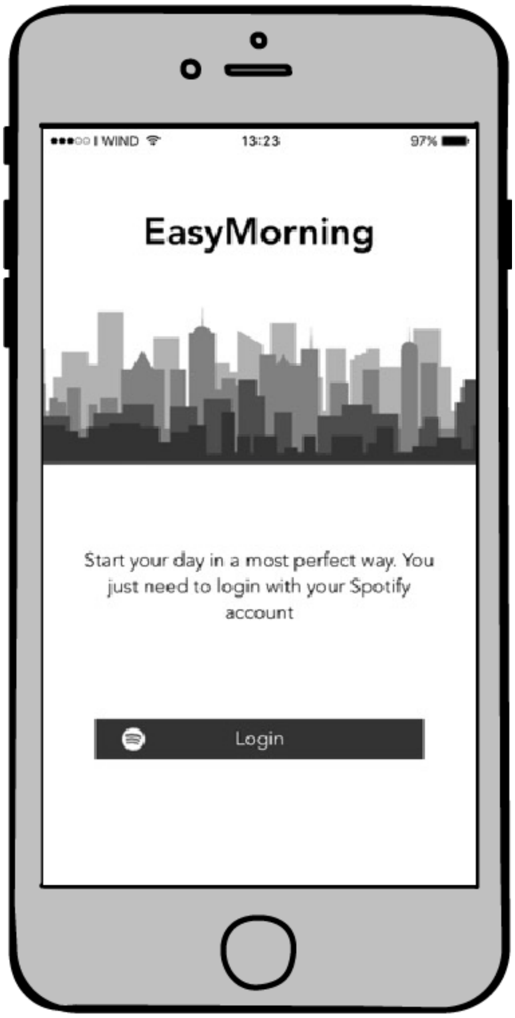

Article
0..n
AddNewDelagate
Alarms
WebViewController
+url: String
-@IBOutlet weak var webView: UIWebView
-@IBAction func cancelPressed(_ sender: UIBarButtonItem)
NewsViewController
+ articles: Array
+ getNews: Bool=false
+ urlRequest: String
+ task: String
-(void)fetchArticles(Data)ResponseData
AlarmGridViewController
+cellWidth =CGFloat(100): Integer
+cellHeight =CGFloat(180): Integer
+alarms: Array
+selectedRow = 0: integer
+Date: NSDate
-(void)displayGridViewAlarms
-(void)longPressed(recognizer: UIGestureRecognizer)
-(void)addNewAlarm
-(void)prepareForSeque(for segue: UIStoryboardSegue, sender: Any
-(void)userNotificationCenter: didReceiveResponse
DetailTabBarController
+alarmToDisplay: Array
+newsVC
+weatherVC
+musicVC
-(void)displayAlarms(Array)
SPTAudioStreamingPlaybackDelegate
AddNewViewController
+snoozeSwitch: UISwitch
+newsSwitch: UISwitch
+spotifySwitch: UISwitch
+forecastForCity: UITextField
+forecastSwitch: UISwitch
+cityStack: UIStackView
+delegate: AddNewDelegate
-(Bool)textFieldShouldReturn
-textFieldDidBeginEditing()
-forecastSwitchPressed()
-@IBAction savePressed(_ sender: UIBarButtonItem)
SPTAudioStreamingDelegate
SpotifyViewController
+spotifyFlag: Bool
+player: SPTAudioStreamingController
+auth = SPTAuth.defaultInstance()
+session: SPTSession
+spotifyDelegateNews
+spotifyDelegateWeather
-(Bool)viewDidAppear
-startPlayingMusic()
- @IBAction func cancelPressed(_ sender: Any)
-initializePlayer(authSession: SPTSession)
-audioStreamingDidLogin(_ audioStreaming: SPTAudioStreamingController!)
-audioStreamingDidLogout(_ audioStreaming: SPTAudioStreamingController!)
-@IBAction func forwardPressed(_ sender: UIButton)
-@IBAction func rewindPressed(_ sender: UIButton) {
-@IBAction func playPressed(_ sender: UIButton)
- @IBAction func pausePressed(_ sender: UIButton)
Scheduler
+actionStart = UNNotificationAction
+actionSnooze = UNNotificationAction
+category = UNNotificationCategory
+trigger = UNTimeIntervalNotificationTrigger
+request = UNNotificationRequest
-scheduleNotification(alarm: Alarm)
-descheduleNotification(alarm: Alarm)
-rescheduleNotification(alarm: Alarm)
SpotifyLoginViewController
+auth = SPTAuth.defaultInstance()
+session: SPTSession
+loginUrl: URL?
-(Bool)viewDidAppear()
-(void)prepareForSeque(for segue: UIStoryboardSegue, sender: Any)
-setupSpotifyAuth()
-updateAfterFirstLogin()
-@IBAction func loginPressed(_ sender: UIButton)
WeatherViewController
+getForecast: Bool
+forecastForCity: String
+currentLocation: CLLocation
-degree: int
-condition: String
-presure: int
-humidity: int
-wind: int
-getCurrentLocationAndData()
-fetchedWeatherDataCurrent()
+weatherData

4 Graphical user interface

This chapter will show the final design and look of the application. And also will be guide for how the application should be used.

4.1 Log in

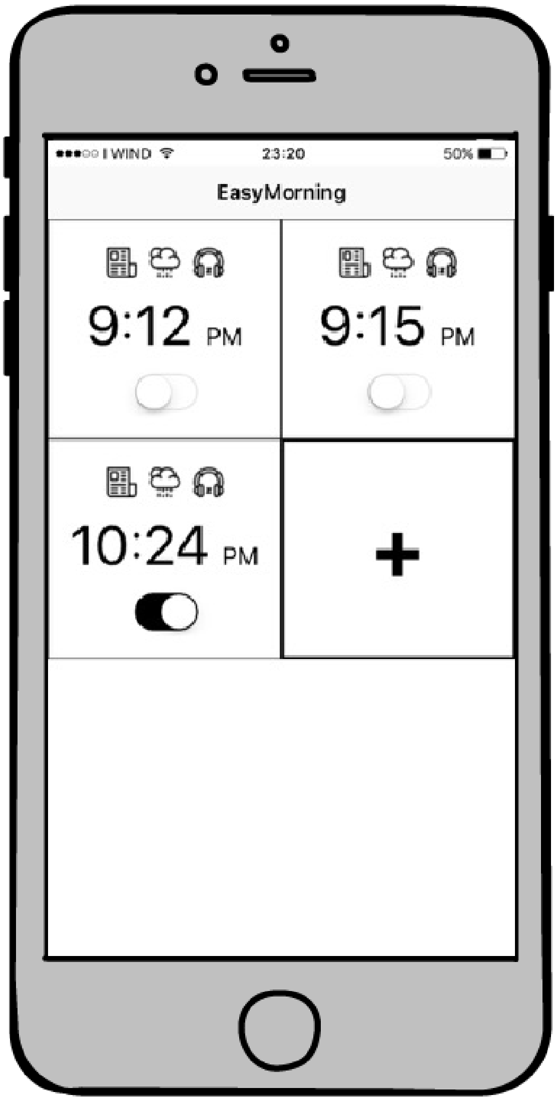
When the user first time opens the application it will be shown short splash screen from our logo and after that he will be redirect to log in to Spotify. You can see the steps and the design on the showed mockup pictures below.



Start your day in a most perfect way. You just need to login with your Spotify account

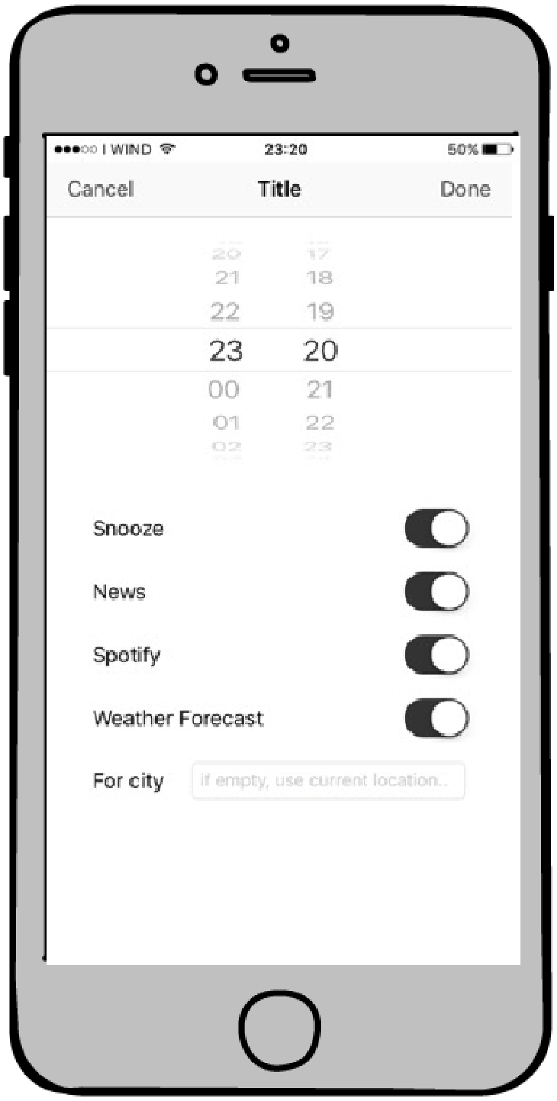
4.2 Main view

Main view of application is the collection view where the alarms are displayed in a grid. Each cell represents an alarm and displays information about it. Information displayed are: time of the alarm trigger, icon showing whether a user choose to display news, weather forecast and Spotify music. It also contains a switch used to enable or disable alarm. In order to delete an alarm from the grid user needs to long press on the desired cell and he will be asked to confirm the deletion. In order if the user wants to add new alarm he needs to click on the "+" button.



4.3 Add new

If the user clicks on the “+” button it will appear new window where the user can add new alarm and select preferences that will be shown when the alarm will be triggered. He can choose between different features that he can select them by tapping on the button next to the name of the feature. He can choose between news, Spotify and weather forecast or he can choose all of them. If he choose weather forecast and he wants weather forecast for the current location he just needs to enable the weather forecast and when the alarm is triggered it will be shown the weather information for the current location, if he wants to see weather information for other city he needs to enter the desired city in the text field.



20 17
21 18
22 19
23 20
00 21
01 22
02 23

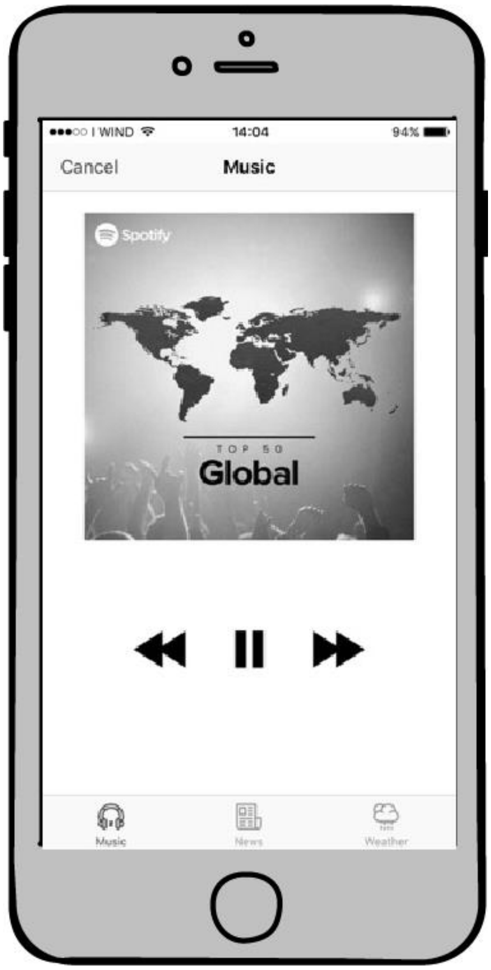
Snooze ●
News ●
Spotify ●
Weather Forecast ●
For city: if empty, use current location..

4.4 Details

Detail view of the alarm is a tabbed view with three components: Spotify music control, News table and Weather forecast. If the user, in times when he/she created the alarm didn't select some of the features to view it (News, Spotify, Forecast switches in the add new alarm), corresponding screen will display the message stating that the screen is unavailable. User can scroll through three screens or go back to the main screen by clicking Cancel.

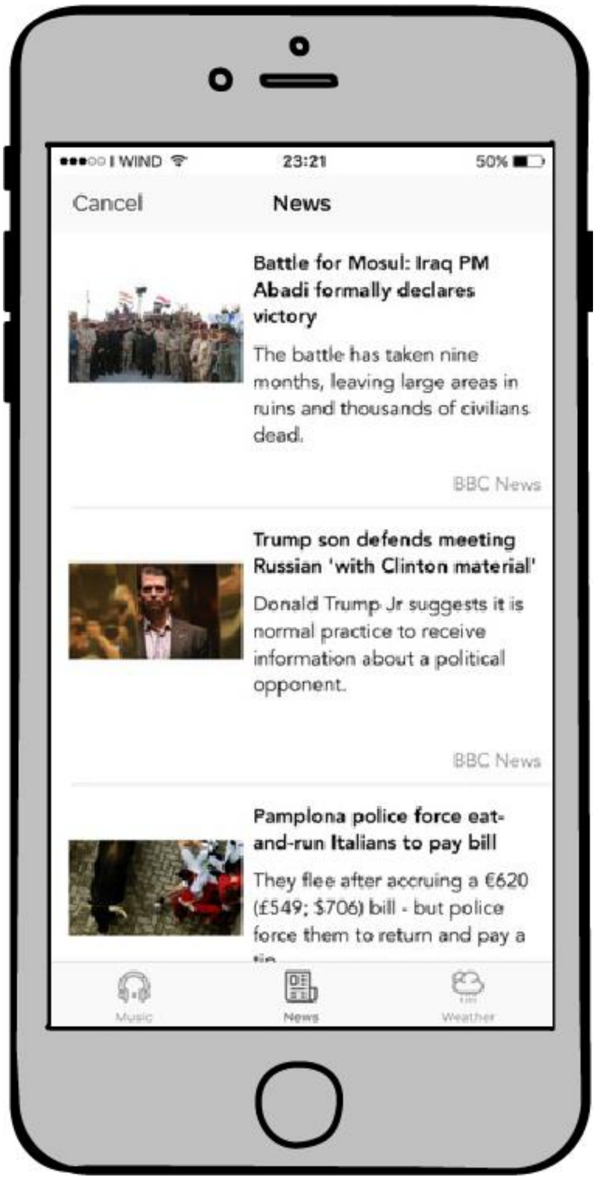
4.5 Spotify view

Here user has audio controls which control the playback of the audio playlist. It can be paused, skipped to the next song or previous one. Music is played from the top charts playlist from Spotify. This will be the first screen shown in details view.



4.6 News view

Table containing top articles for today from BBC rss feed. It displays the headline, small summary and picture. User can click on an article he wants to read and he will be redirected to a new screen containing the whole article. He can always go back by pressing Cancel.



Battle for Mosul: Iraq PM Abadi formally declares victory

The battle has taken nine months, leaving large areas in ruins and thousands of civilians dead.

BBC News

Trump son defends meeting Russian 'with Clinton material'

Donald Trump Jr suggests it is normal practice to receive information about a political opponent.

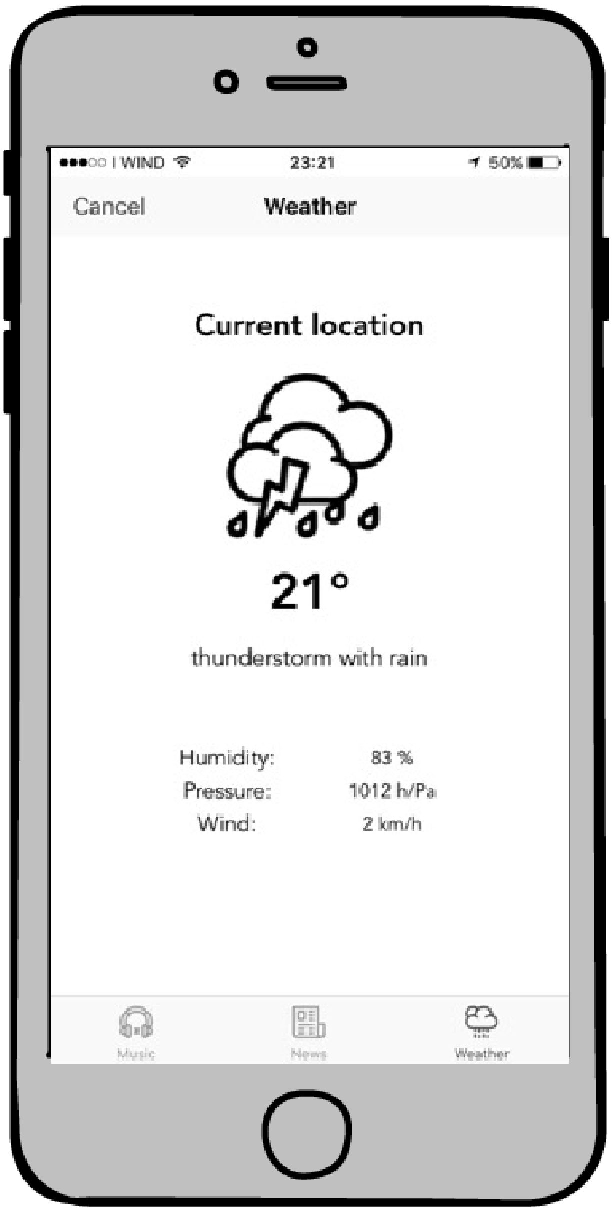
BBC News

Pamplona police force eat-and-run Italians to pay bill

They flee after accruing a €620 (£549; $706) bill - but police force them to return and pay a

4.7 Weather view

Here user can view the information about the current weather. On top of the screen will be the name of the entered city or the current location. Also is shown picture that changes if the weather changes. The user also can see other information about the selected weather information.



Current location

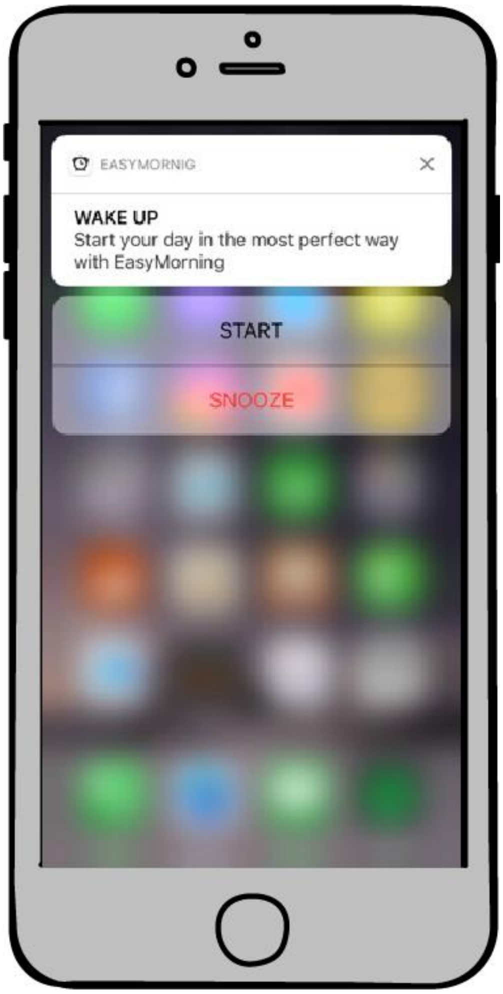
21°

thunderstorm with rain

Humidity:	83 %
Pressure:	1012 h/Pa
Wind:	2 km/h

4.8 Notification

When an alarm fires an notification will be received playing the alarm sound. Notification has two actions which the user can select. START action which cancels the alarm and takes him to details view of the alarm (where music starts playing, news articles and weather forecast are fetched). SNOOZE action will reschedule the alarm for 5 minutes.

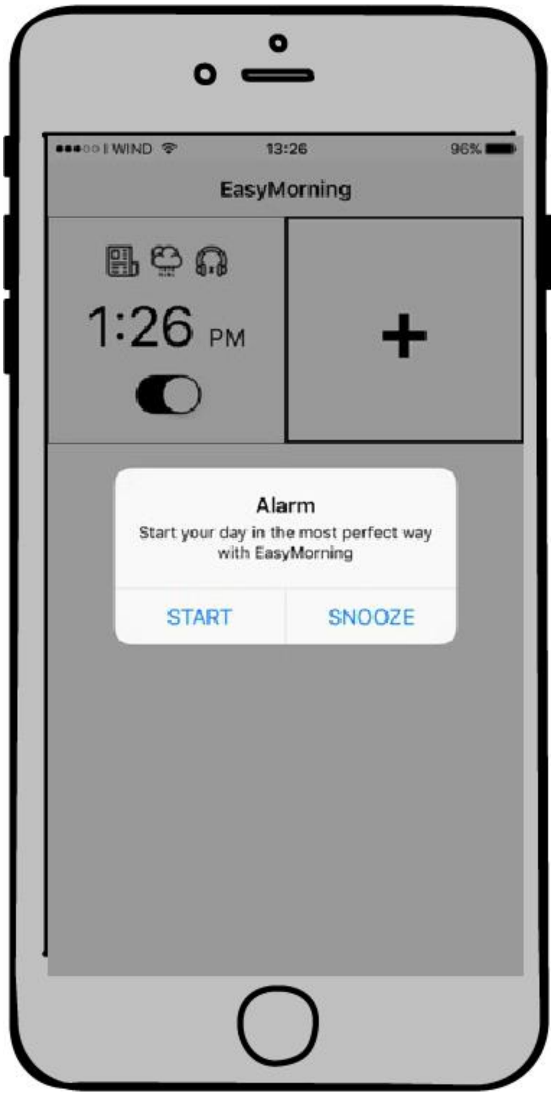


EASYMORNIG

WAKE UP

Start your day in the most perfect way with EasyMorning

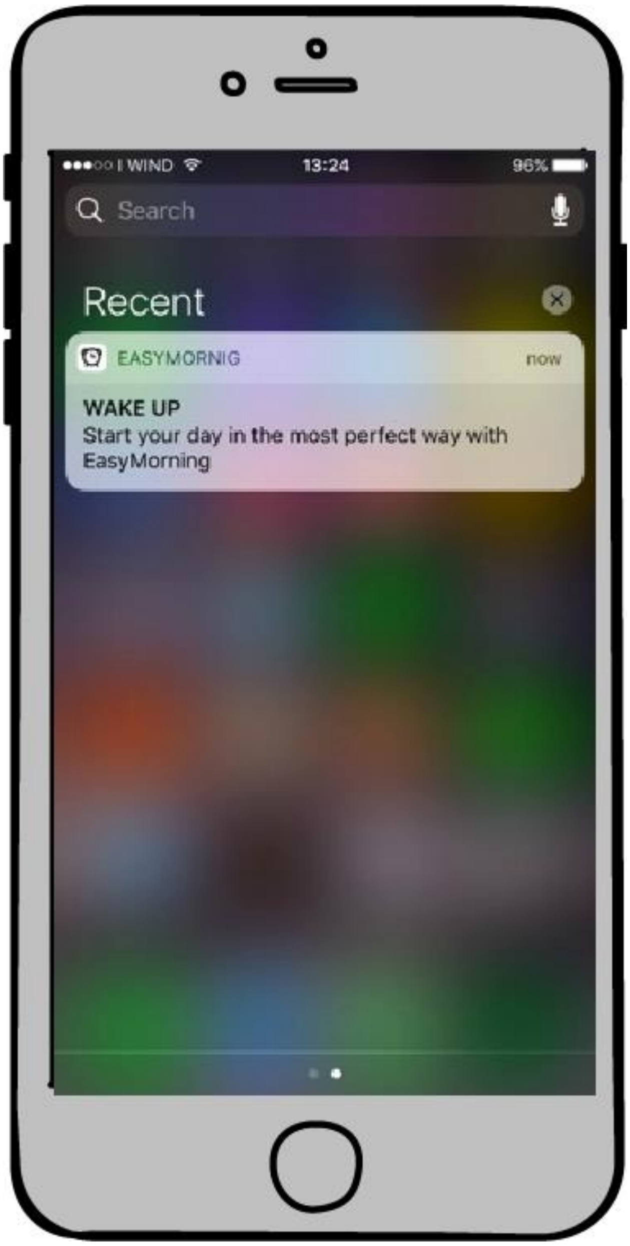
If the user is in the app, notification will not be fired but there will be an alert box displaying same actions as notification.



Alarm

Start your day in the most perfect way with EasyMorning

If the user is in the notification center, notification will be there until you select some of the previous defined actions.



WAKE UP
Start your day in the most perfect way with EasyMorning

5. References

[1] Design and Implementation of Mobile Applications course material [Online], http://home.deib.polimi.it/baresi/dima.htm
[2]IOS developer library documentation, https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/index.html
[3]Spotify SDK documentation, https://developer.spotify.com/technologies/spotify-ios-sdk/