

Requirements and Analysis Document (RAD) for Group 14

Version: 1.0

Date: 2015-05-31

Authors: Johan Ben Mohammad, Erik Forsberg, Patrick Franz, Michael Tran

This version overrides all previous versions.

1 Introduction

This is a simple and easy-to-use desktop calendar. It incorporates a schedule with activities and a contacts section where you can store contact information to other people.

1.1 Purpose of application

To simplify planning and help people overview their activities. At work and/or at home.

1.2 General characteristics of application

The application will be a desktop application working for Windows/Mac/Linux platforms. It is intended to be standalone (non-networked) for now, but might be extended with the possibility to synchronize local data to a server and vice versa (i.e. get the latest data from a server database). The user should be able to add activities to a calendar and get a clear overview on upcoming activities.

1.3 Scope of application

Desktop application for one user, but works with multiple users if there are multiple program files in different folders. Saving data after shutting down the application is crucial.

1.4 Objectives and success criteria of the project

The goal is to make the calendar as easy to use and as intuitive as possible. It must also have some necessary basic functionality to be able to create a personal schedule and contact book. The application should be runnable and work on any of the platforms using a graphical user interface.

1.5 Definitions, acronyms and abbreviations

- GUI, graphical user interface.
- Java, platform independent programming language.
- JRE, the Java Runtime Environment. Additional software needed to run a Java application.

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

1. Add contact
2. Edit contact
3. Remove contact
4. Add activity
5. Edit activity
6. Remove activity
7. Change view
8. Add contact to a group
9. Remove contact from a group
10. Edit group settings (name etc.)
11. Remove group
12. Add group
13. Change color theme (in settings)
14. Change default view (in settings)
15. Add notification (for an activity)
16. Edit notification (for an activity)
17. Remove notification (for an activity)
18. Receive notification
19. Change date/time format (in settings)
20. General search for activities/contacts/groups
21. Availability to hide and show the menu bar

2.2 Non-functional requirements

2.2.1 Usability

The calendar should be easy to use and as intuitive as possible. This is the main goal for this application. The graphic user interface (GUI) should be easy to understand. Any user with previous experience with digital calendars and/or other desktop applications should be able to recognize functions and their intended use without any problems.

2.2.2 Reliability

N/A

2.2.3 Performance

Performance will not have a high priority in this application, although the aim is to perform all types of operations without any significant delay (1-2 seconds in worst case scenario).

2.2.4 Supportability

The application is developed to be desktop only. However separation of the GUI will be prioritized to be able to adapt the application with a new GUI for different platforms (Phones, tablets etc.).

As the application uses a database to store data, the application can be modified to use a client/server structure where a user would be able to use a login system and retrieve their data from any computer with the application installed.

2.2.5 Implementation

Java will be used to implement the application. Any computer must have the Java Runtime Environment (JRE) to be able to run the application.

2.2.6 Packaging and installation

The application will be packaged as a .JAR file together with any external resources such as images. To be able to run the application, the user must have these resources.

2.2.7 Legal

As this is a school project, no legal documentation is needed.

2.3 Application models

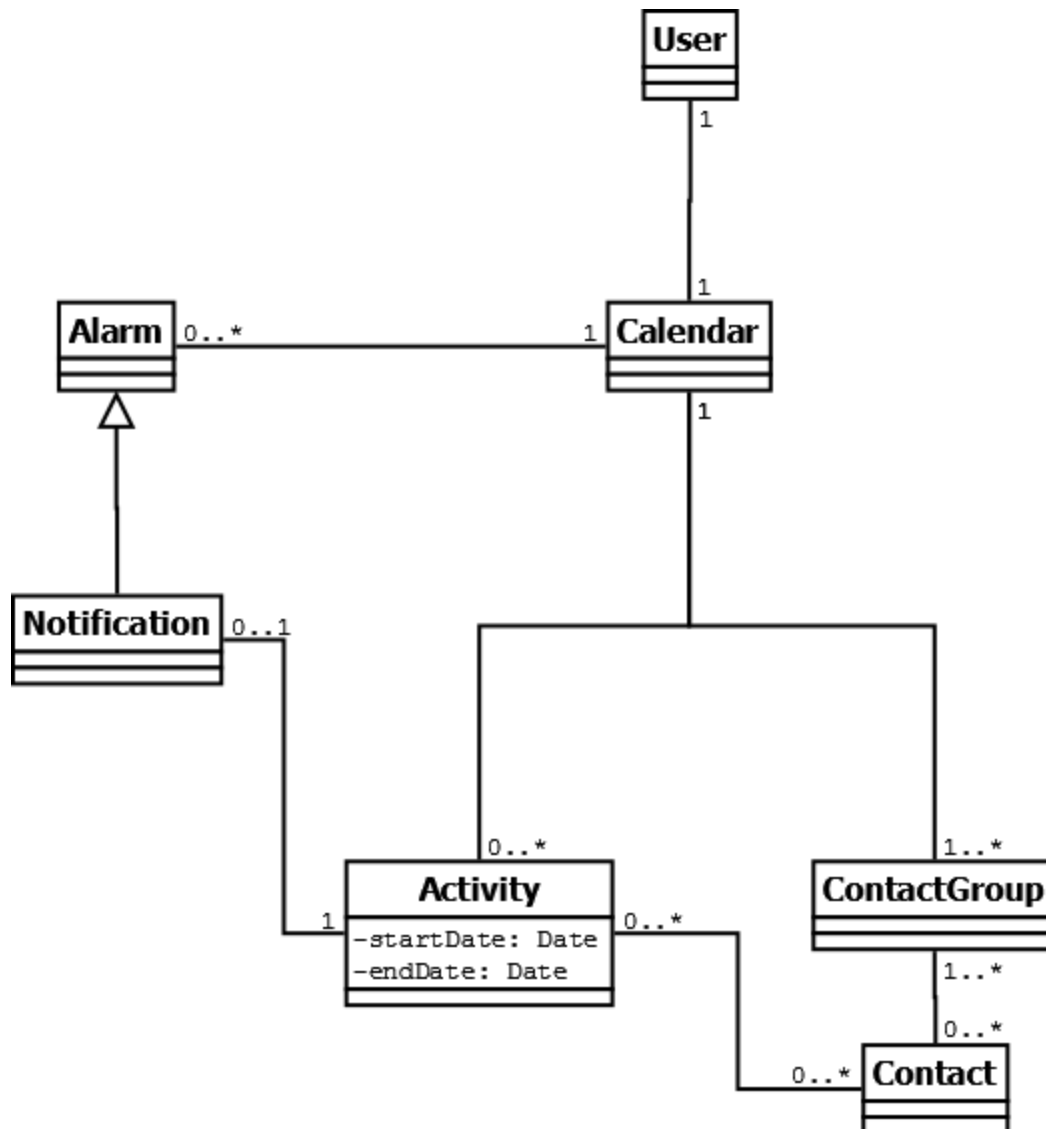
2.3.1 Use case model

N/A

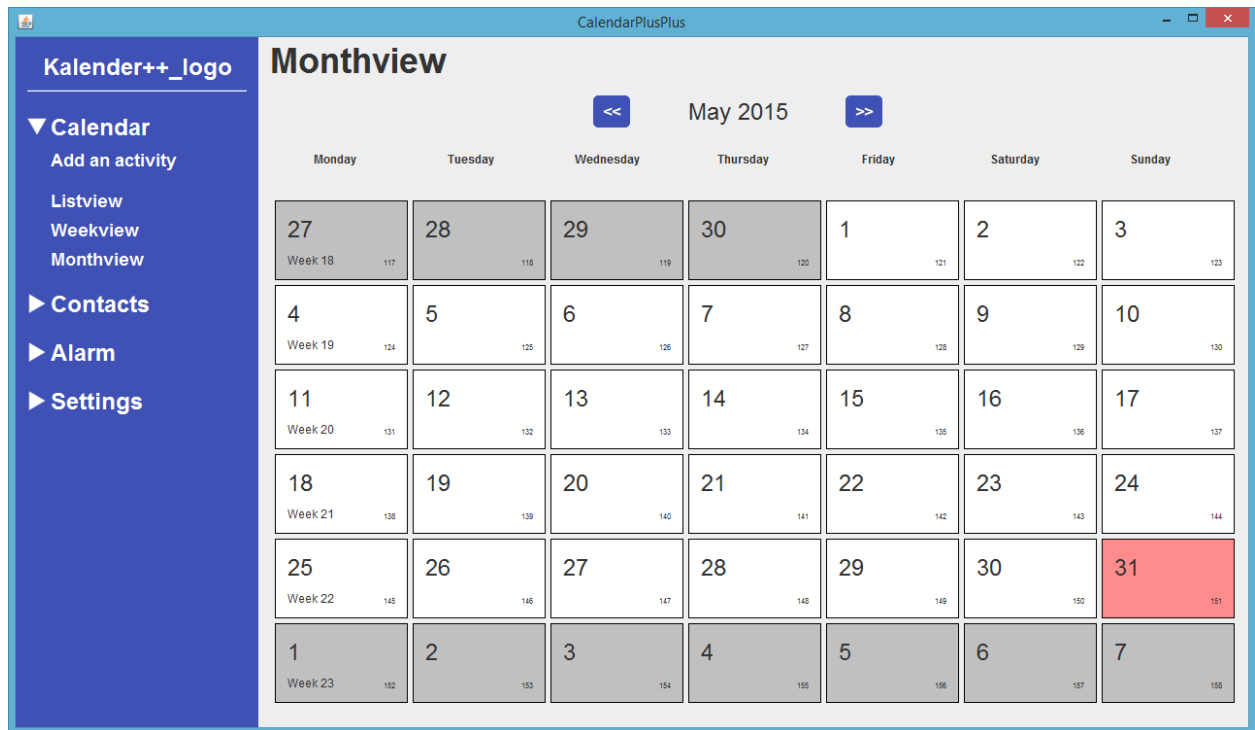
2.3.2 Use cases priority

See 2.1

2.3.3 Domain model



2.3.4 User interface



2.4 References

Use case texts:

Use case: Add activity

Priority: 1

Normal flow

User	System
1: Clicks the “Add an activity” button	
	2: Displays “add an activity” form.
3: Fills in required fields.	
4: Clicks save button.	
	5: Displays overview of the added activity.

Alternative flow: Required fields missing

User	System
	3.1: Prompt user to fill in data in empty required fields.
3.2: Fills in data in empty required fields. GOTO: 4.	

Use case: Add contact

Priority: 2

Normal Flow

User	System
1: Clicks "Add new contact" button.	
	2: Displays "Add a contact" form.
3: Fills in required fields.	
4: Clicks save button.	
	5: Displays list of all contacts.

Alternative flow: Required fields missing

User	System
	3.1: Prompt user to fill in data in empty required fields.
3.2: Fill in data in empty required fields. GOTO: 4.	

Use case: Edit activity

Priority: 3

Normal Flow

User	System
1: Clicks "Edit activity" button.	
	2: Displays "Edit activity" form.
3: Edit fields.	
4: Clicks save button.	
	5: Displays overview of the edited activity.

Alternative flow: Required fields missing

User	System
	3.1: Prompt user to fill in data in empty required fields.
3.2: Fill in data in empty required fields. GOTO: 4.	

Use case: Change calendar view (week/month)

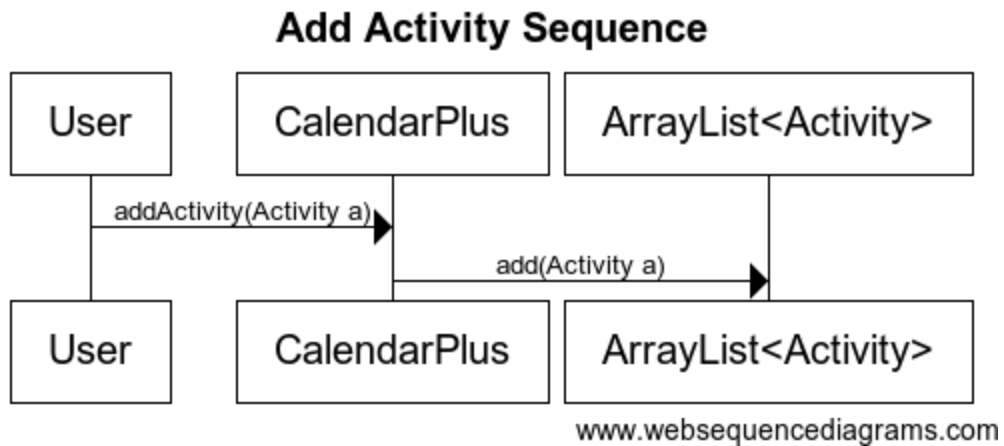
Priority: 4

Normal Flow

User	System
1: Click on alternate view button	
	2: Change current calendar view.
	3: Disable current view button.
	4: Enable previous view button.

Sequence Diagrams:

Add activity:



Add contact:

