

1 Vision

The purpose of this program is to make a calendar that gives a graphical overview of the daily tasks.

2 Use Cases

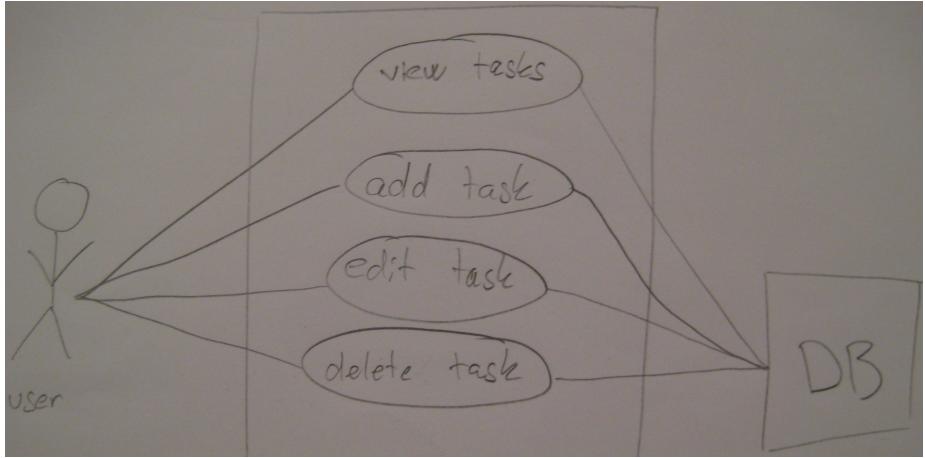
- UC1: Manipulating information

- **Primary actor:**
The user
- **Stakeholders and interests:**
The user: Wants an overview of the daily or weekly tasks
- **Precondition:**
The user is logged in to the system
- **Postcondition:**
The changes the user made is saved in the system.
- **Basic flow:**
The user logs in to the system and either
 - * a: types in a task
 - * b: edit an existing task
 - * c: deletes an existing task

- UC2: Viewing information

- **Primary actor:**
The user
- **Stakeholders and interests:**
The user: Wants an overview of the daily or weekly tasks
- **Precondition:**
The user is logged in to the system
- **Postcondition:**
No changes has been made to any tasks.
- **Basic flow:**
The user logs in to the system and either
 - * a: view already existing tasks, daily
 - * b: view already existing tasks, weekly

3 Use Case UML Diagrams



4 Glossary

Task - Is used to describe any event that is added to the calendar

5 Supplementary Requirements

The program must...

- **Functionality**

support the tasks that a user needs in order to sustain an up to date calendar, that can be used to view task throughout the weeks. Furthermore it should not be possible for different users to view other users tasks.

- **Usability**

be easy to use, enabling almost anyone to use it without much hassle.

- **Reliability**

be available at all times, as the need for the calendar can emerge at any given time.

- **Performance**

not use unnecessary resources and should be able to show tasks for a given period within seconds of request.

- **Supportability**

be maintainable by other programmers than the original. Additionally it must be possible to extend the program with functionality. The program should be able to run on most modern devices.