

# 1 Glossery

**Event** Something predicted to happen at some point.

# 2 Vision

The purpose of this product is to make browsing, management and sharing easy for people with little or no knowledge about interaction with networks or computers.

# 3 User cases

**Name** View calender

**Scope** Calender client

**Level** User goal

**Primary Actor** User

**Stakeholders and Interests** User

**Precondition** User is connected to database. User has logged on.

**Postcondition**

**Main Success Scenario** User gets the desired information.

**Extensions** No database connection returns error

**Special Requirements**

**Name** Create event

**Scope** Calender client

**Level** User goal

**Primary Actor** User

**Stakeholders and Interests** User

**Precondition** User is connected to database. User has logged on.

**Postcondition** An event has been created in the database and the view is updated

**Main Success Scenario** User creates the desired event

**Extensions** No database connection returns error

**Special Requirements**

**Name** Delete event

**Scope** Calender client

**Level** User goal

**Primary Actor** User

**Stakeholders and Interests** User

**Precondition** User is connected to database. User has logged on. User has chosen a single event in the users personal calender.

**Postcondition** the chosen event has been deleted in the database and the view is updated.

**Main Success Scenario** User deletes the desired event

**Extensions** No database connection returns error

**Special Requirements**

**Name** Edit event  
**Scope** Calender client  
**Level** User goal  
**Primary Actor** User  
**Stakeholders and Interests** User  
**Precondition** User is connected to database. User has logged on. User has chosen a single event in the users personal calender.  
**Postcondition** The chosen event has been updated in the database and the view is updated  
**Main Success Scenario** User updates the desired event  
**Extensions** No database connection returns error  
**Special Requirements**

**Name** User login  
**Scope** Calender client  
**Level** User goal  
**Primary Actor** User  
**Stakeholders and Interests** User  
**Precondition** User is connected to database.  
**Postcondition** The user has successfully logged on to the system  
**Main Success Scenario** User logs on to the system  
**Extensions** No database connection returns error Wrong e-mail address and/or passwords returns error and offers an opportunity to get the password sent by mail  
**Special Requirements**

**Name** Share calender  
**Scope** Calender client  
**Level** User goal  
**Primary Actor** User  
**Stakeholders and Interests** User  
**Precondition** User is connected to database. User is logged on  
**Postcondition** The user has sent a share-invite to another user of the system  
**Main Success Scenario** The user successfully sends a share-invite to another user  
**Extensions** No database connection returns error The entered e-mail which the user wants to share the calender does not exist in the system. The user is presented with an error  
**Special Requirements**

**Name** Accept request  
**Scope** Calender client  
**Level** User goal  
**Primary Actor** User  
**Stakeholders and Interests** User  
**Precondition** User is connected to database. User is logged on User has received an invite to see another users calender  
**Postcondition** The user has accepted the invite and is now able to see the shared calender. The view is updated.  
**Main Success Scenario** The user has access to see the shared calender

**Extensions** No database connection returns error The user declines the invite, nothing happens

**Special Requirements**

## 4 Supplementary specification

The supplementary specification is structured using the FURPS+ model.

**Functionality:** The program should support a minimum of functions.

- The functions should support sharing of calendars, creation and modification of events and viewing.
- The functions should be intuitively grouped.

**Usability:** The user should be able to use the program with as little knowledge as possible, which raises the following requirements:

- The user should consider a minimum of options performing the desired task. This may compromise both performance and functionality.
- Anyone beyond the age of 12 should be able to understand the program terminology.

**Reliability:** The program should be very reliable, even if it compromises security. Any error should result in recovery or crash, neither involving the user. Any crash or recovery must maintain the content of the database.

- The program should be able to complete as many tasks as possible without internet connection.

**Performance:** The program should have reasonable performance. Since it does not rely on any time expensive operations, no supplementation is needed.

**Supportability:** Functions should be supported with a help page. The content should be limited because the intuitiveness and limited functionality makes it needless.