

# Software Requirements Document

## Introduction

### Purpose

The goal is to create an add-on for the software VANTED (<http://vanted.org>) that implements fast stress minimization and a multilevel graph drawing framework respectively. The add-on is to be developed for the class “Softwareprojekt” in the summer term 2019 at the University of Konstanz. This document lays out the requirements for the add-on.

### Scope

As already mentioned, the software is to be developed as add-on for the VANTED system. The software shall be developed in the Java programming language. Two network layout methods for undirected graphs are to be added.

The stress minimization approach which tries to minimize a so-called stress function and the multilevel framework which is not an actual layout algorithm but a way to apply another layout algorithm at each level of a ‘hierarchy’ of simplified versions of the graph.

### Abbreviations, Acronyms and Definitions

- **VANTED** refers to “Visualisation and Analysis of Networks containing Experimental Data”.
- **Add-on** refers to an extension of the VANTED software that can be loaded by VANTED and enhances its functionality.
- **OGDF** refers to “Open Graph Drawing Framework”

### References

- **[Spec]** Softwareprojekt2019\_Gruppe2Thema.pdf
- **[V]** VANTED source code (<https://bitbucket.org/vanted-dev/vanted/src/master/>)
- **[Meet]** The initial meeting on the 25 April 2019

### Overview

This document is divided into three chapters. The first chapter (which you are reading right now) gives an introduction and defines the purpose and scope of the software product. It also lists abbreviations and lists relevant definitions. The second chapter is a non-formalized overview of the software that is to be developed. The third chapter lists specific requirements according to the IEEE 830-1998 standard illustrates the system by means of a use-case diagram.

# Overall description

## Product Perspective

The add-on can be used in the same way as existing layouters within VANTED.  
The software shall function as specified by the requirements and shall be thoroughly tested and documented.

## Product Functions

The add-on enables users to visualize graphs/networks in a visually pleasing way. It provides a settings dialogue where the user can set parameters for the layout methods. The usage of the add-on shall not require sophisticated knowledge of the underlying methods.

## User Characteristics

The target group are users of the VANTED software.

## Constraints

The stress minimization shall be fast (as perceived by Karsten Klein). The software should be finished until 7 July 2019.

## Assumptions and Dependencies

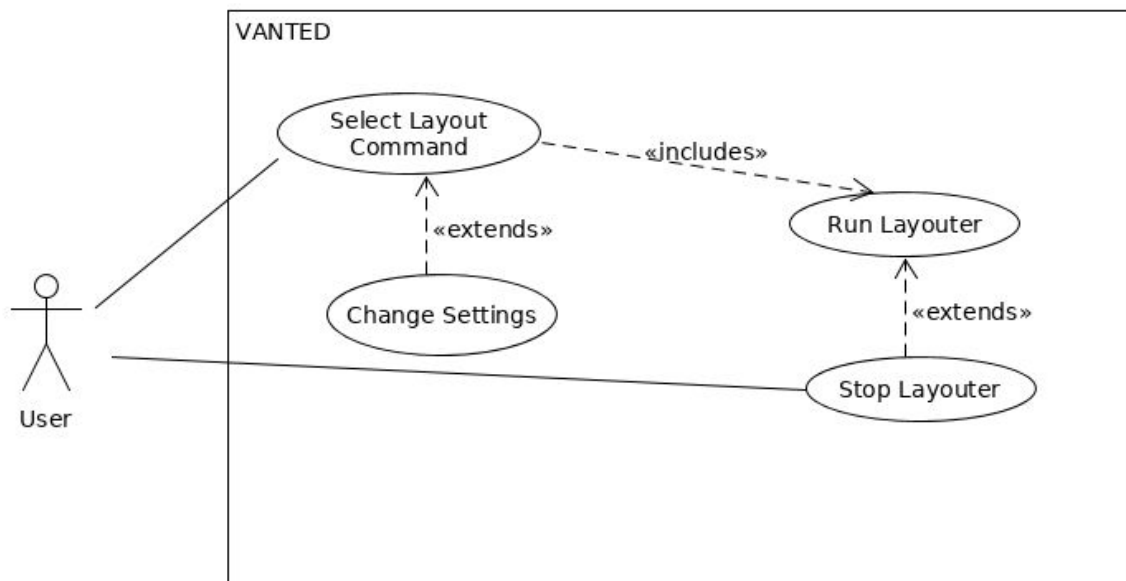
Users are already able to use the VANTED software and possess a sufficiently powerful computer to comply with the memory and processing requirements of the implemented methods.

## Specific Requirements

### External Interfaces

The usage shall be similar to existing layouters already implemented in VANTED.

## Use Case Diagram



Both layout methods will be used in the same way. First, the user selects the Layout Command in the “Layout” menu in VANTED. Then they can optionally change the settings and click on “Layout Network”. If the layouting process takes too long, they can click on “Stop Layout” to stop it.

## Functions

### R0: LayoutCommandStressMinimization

Function: The add-on shall add a layout command that performs stress minimization.

Description: The layout command shall be selectable from the layout tab in VANTED.

Source: [Spec]

Dependency: –

### R1: LayoutCommandMultilevelFramework

Function: The add-on shall add a layout command that uses a multilevel approach for layouting.

Description: The layout command shall be selectable from the layout tab in VANTED.

Source: [Spec]

Dependency: –

### R2: KindsOfGraphs

Function: The two layout commands shall work on undirected graphs without edge weights only.

Description: We investigated the Graph and Edge data structures in VANTED and there doesn't appear to be a way to set edge weights.

Source: [Spec], [V]

Dependency: R0, R1

### R3: OutputDimension

Function: The resulting layout shall be displayed in two dimensions.

Description: –

Source: –

Dependency: –

### R4: DistanceMeasurement

Function: Distances in the output shall be measured in Euclidean space.

Description: This is important for the stress minimization function.

Source: –

Dependency: –

### R5: MultilevelFrameworkAlgorithmSelection

Function: It shall be possible to change which algorithm the multilevel framework uses on each level.

Description: The specified algorithm will then be used for all levels. It shall not be possible to use different algorithms for different levels in the same execution of the multilevel layouter.

Source: [Spec]

Dependency: R1

### R6: Efficiency

Function: The layouters shall be implemented efficiently.

Description: They shall be able to calculate layouts for large graphs (several thousand vertices) not significantly slower than existing implementations in OGDF.

Source: [Spec]

Dependency: –

### R7: JavaDoc

Function: All nontrivial implemented methods shall be documented using JavaDoc.

Description: Methods such as getters and setters count as trivial and need not be documented if they exhibit their expected behavior.

Source: [Spec]

Dependency: –

### R8: UserManual

Function: A short PDF user manual describing the usage of the layouters shall be written.

Description: The user manual shall describe briefly what all elements (e.g. buttons, settings) that the add-on adds to VANTED do.

Source: [Spec]

Dependency: –

### R9: StressMinimizationSettings

Function: The behavior of the stress minimization layouter shall be changable using settings.

Description: –

Source: R0

Dependency: –

#### R10: UnitTests

Function: Unit tests shall be applied where reasonable.

Description: Most non-trivial methods should have a corresponding unit test.

Source: [Spec]

Dependency: –

#### R11: Benchmarks

Function: TODO

Description: TODO

Source: [Spec]

Dependency: –

#### R12: ChangeSubAlgorithmParameters

Function: The parameters of the algorithm that the multilevel framework uses on all the levels shall be changeable.

Description: The settings interface of the algorithm shall be shown before the multilevel framework is run, allowing the user to change the parameters.

Source: –

Dependency: R1, R5

#### R13: MultilevelFrameworkCoarseningInterface

Function: There shall be an interface allowing the addition of further coarsening methods.

Description: It shall be possible to add new coarsening methods without changing the implementation of the multilevel framework itself.

Source: [Meet]

Dependency: R1

#### R14: MultilevelFrameworkPlacementInterface

Function: There shall be an interface allowing the addition of further placement methods.

Description: It shall be possible to add new placement methods without changing the implementation of the multilevel framework itself.

Source: –

Dependency: R1

#### R15: WorkOnSelection

Function: If nodes are selected the framework and algorithm shall only work with these nodes.

Description: It shall be possible for the user to use the selection function of VANTED to select a specific subset of the nodes and let the framework and algorithm only operate on these nodes while not moving the other nodes.

Source: [V]

Dependency: –

#### R16: JavaVersion

Function: The add-on shall be developed and tested using Java version 8.

Description: –

Source: –

Dependency: –

## Project time schedule

Week/Final Date	Event/Tasks
29.4. - 5.5.	Read papers, Get development environment working
6.5. - 12.5.	Finish requirements document
13.5. - 19.5.	Decide on design, Write design specification
19.5.	Milestone 1: <ul style="list-style-type: none"><li>• Design specification written</li></ul>
20.5. - 26.5.	
27.5. - 2.6.	
3.6. - 9.6.	
9.6.	Milestone 2: <ul style="list-style-type: none"><li>• First prototype working</li></ul>
10.6. - 16.6.	
17.6. - 23.6.	
24.6. - 30.6.	
1.7. - 7.7.	
7.7.	Milestone 3: <ul style="list-style-type: none"><li>• Finished product</li><li>• Pass tests and benchmarks</li></ul>
8.7. - 14.7.	Polish project
15.7. - 21.7.	Time buffer, create final presentation
22.7. - 23.7.	Practise final presentation
23.7.	Final presentation