

# System design document for NN

Version:

Date:

Author:

This version overrides all previous versions.

## 1 Introduction

General info. What is this? What does it describe?

### 1.1 Definitions, acronyms and abbreviation

Some definitions etc. probably same as in RAD

## 2 System architecture

The most overall, top level, description of the system. Which (how many) machines are involved? What software on each (which versions). Which responsibility for each software? Dependency analysis. If more machines: How do they communicate? Describe the high level overall flow of some use case. How to start/stop system.

An [UML deployment diagram](#), possibly drawings and other explanations.  
Possibly UML [sequence diagrams](#) for flow.

(Persistence and Access control further down)

Any general principles in application? Flow, creations, ...

## 3. Subsystem decomposition

For each identified software above (that we have implemented), describe it ...

### 3.1 "...First software to describe" ...

Recap: What is this doing (more detailed)

Divide it into top level subsystems. An [UML package diagram](#) for the top level. Describe responsibilities for each package (subsystem). Describe interface. Describe the flow of some

use case inside this software. Try to identify abstraction layers. Dependency analysis  
Concurrency issues.

If a standalone application

- Here you describe how MVC is implemented
- Here you describe your design model (which should be in one package and build on the domain model)
- A class diagram for the design model.

else

- MVC and domain model described at System Architecture

Diagrams

- Dependencies ([STAN](#) or similar)
- UML sequence diagrams for flow.

Quality

- List of tests (or description where to find the test)
- Quality tool reports, like [PMD](#) (known issues listed here)

NOTE: Each Java, XML, etc. file should have a header comment: Author, responsibility, used by.., uses ...

## 3.2 “...next software to describe” ...

As above....

## 4. Persistent data management

How does the application store data (handle resources, icons, images, audio, ...). When?  
How? URLs, paths, ... data formats... naming..

## 5. Access control and security

Different roles using the application (admin, user, ...)? How is this handled?

## 6. References