

Guidelines for the Design Documentation

Fritz Solms

March 5, 2014

The design document needs to cover both,

- the software architecture design addressing the non-functional requirements and
- the application design covering the non-functional requirements.

1 Software architecture design

The software architecture design is done subject to the architecture constraints as specified in requirements specification. It must include a justification of the architectural decisions based on quality requirements

- choices of technologies (e.g. programming languages, database technologies, application servers, ...),
- chosen frameworks (e.g. object/relational mappers, web services frameworks),
- chosen protocols (e.g. JSON vs XML), and
- chosen libraries (e.g. for generating PDFs, JSON or XML marshalling/demarshalling, LDAP integration ...).

This section needs to demonstrate how the system will be able, within designed software architecture, to address the non-functional requirements.

2 Application design

Application design must address functional requirements as specified in requirements specification. Whilst the functional/application requirements focused on what the system needs to do from a user's perspective, the design document focuses on how the system addresses the functional requirements.

You should be guided by design principles like separation of concerns (single responsibility principle), low cohesion and coupling, Use design

patterns where applicable — they can significantly improve the flexibility and quality of your design.

You should separate the design of

1. the back-end system containing the main functionality of the system from,
2. the web application providing a browser-based user interface into the application,
3. the Android application

Each of these will have APIs, classes and processes. In particular, you should separate the user interface processes which assemble the information to make a service request to the application backend from the backend processes realizing those services.

The application design should include

- the specifications of lower levels of granularity,
- API specifications in the form of UML interfaces and class diagrams for the inputs and outputs and a separate exception class for each pre-condition,
- class diagrams with system classes with attributes (including types), methods, relationships
- sequence and/or activity diagrams for detailed system process specifications,
- UI screen designs and user work-flow specifications (activity or sequence diagrams),
- the design of the database tables, ...