**Purpose**

System design is documented in the System Design Document (SDD). It describes design goals set by the project, subsystem decomposition (with UML class diagrams), hardware/software mapping (with UML deployment diagrams), data management, access control, control flow mechanisms, and boundary conditions. The SDD is used to define interfaces between teams of developers and serve as a reference when architecture-level decisions need to be revisited.

**Audience**

The audience for the SDD includes the project management, the system architects (i.e., the developers who participate in the system design), and the developers who design and implement each subsystem.

**Template**

|  |  |
| --- | --- |
| **Outline** | **Description** |
| **1. Introduction** 1.1 Purpose of the system 1.2 Design goals 1.3 Definitions, acronyms, and abbreviations 1.4 References 1.5 Overview | *The purpose of the Introduction is to provide a brief overview of the software architecture and the design goals. It also provides references to other documents and traceability information (e.g., related requirements analysis document, references to existing systems, constraints impacting the software architecture).* |
|  | |
| **2. Current software architecture** | *The second section describes the architecture of the system being replaced. If there is no previous system, this section can be replaced by a survey of current architectures for similar systems. The purpose of this section is to make explicit the background information that system architects used, their assumptions, and common issues the new system will address.* |
|  | |
| **3. Proposed software architecture** | *The third section documents the system design model of the new system.* |
|  | |
| **3.1 Overview** | *The overview presents a birds-eye view of the software architecture and briefly describes the assignment of functionality to each subsystem.* |
|  | |
| **3.2 Subsystem decomposition** | *Subsystem decomposition describes the decomposition into subsystems and the responsibilities of each. This is the main product of system design.* |
|  | |
| **3.3 Hardware/software mapping** | *Hardware/software mapping describes how subsystems are assigned to hardware and off-the-shelf components. It also lists the issues introduced by multiple nodes and software reuse.* |
|  | |
| **3.4 Persistent data management** | *Persistent data management describes the persistent data stored by the system and the data management infrastructure required for it. This section typically includes the description of data schemes, the selection of a database, and the description of the encapsulation of the database.* |
|  | |
| **3.5 Access control and security** | *Access control and security describes the user model of the system in terms of anaccess matrix. This section also describes security issues, such as the selection of an authentication mechanism, the use of encryption, and the management of keys.* |
|  | |
| **3.6 Global software control** | *Global software control describes how the global software control is implemented. In particular, this section should describe how requests are initiated and how subsystems synchronize. This section should list and address synchronization and concurrency issues.* |
|  | |
| **3.7 Boundary conditions** | *Boundary conditions describes the start-up, shutdown, and error behavior of the system. (If new use cases are discovered for system administration, these should be included in the requirements analysis document, not in this section.)* |
|  | |
| **4. Subsystem services Glossary** | *The fourth section, Subsystem services, describes the services provided by each subsystem in terms of operations. Although this section is usually empty or incomplete in the first versions of the SDD, this section serves as a reference for teams for the boundaries between their subsystems. The interface of each subsystem is derived from this section and detailed in the Object Design Document.* |