**sAttendance Tracker**



Software Design Document

Version: 1.0

11.11.2018

Mertali Köprülü

Metin Berk Karataş

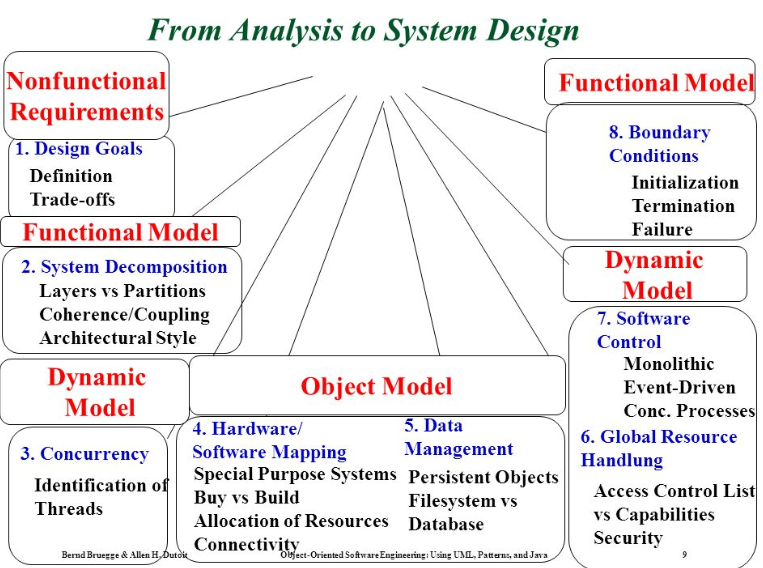
Gülden Güllü

Korhan Köz

Prepared for

SE301 Software Engineering





|  |  |  |  |
| --- | --- | --- | --- |
| Topic: | Name: | Topic: | Name: |
| 1.1.Purpose of the System | + | 3.3. Hardware Software Mapping | (Korhan) |
| 1.2.Design Goal | (Korhan) | 3.4. Persistent Data Management | (DBER)(Mertali) |
| 1.3. Definitions,  Acronyms, and Abbreviations | + | 3.5 Access Control and Security |  |
| 1.4. References | (Korhan) | 3.6. Global Software Control |  |
| 2.Current Software Architecture | (Berk) | 3.7. Boundary Services |  |
| 3.1.  Overview | (Korhan) | 4. Subsystem Services |  |
| 3.2.  System Decomposition |  | 5.References | + |

Table of Contents

[1. Introduction 1](#_Toc433996772)

[1.1. Purpose of the System 1](#_Toc433996773)

[1.2. Design Goals 1](#_Toc433996774)

[1.3. Definitions, Acronyms, and Abbreviations 1](#_Toc433996775)

[1.4. References 1](#_Toc433996776)

[2. Current Software Architecture 1](#_Toc433996777)

[3. Proposed Software Architecture 1](#_Toc433996778)

[3.1. Overview 1](#_Toc433996779)

[3.2. System Decomposition 1](#_Toc433996780)

[3.3. Hardware Software Mapping 2](#_Toc433996781)

[3.4. Persistent Data Management 2](#_Toc433996782)

[3.5. Access Control and Security 2](#_Toc433996783)

[3.6. Global Software Control 2](#_Toc433996784)

[3.7. Boundary Conditions 2](#_Toc433996785)

[4. Subsystem Services 2](#_Toc433996786)

[5. References 2](#_Toc433996787)

SYSTEM DESIGN DOCUMENT[Error: Reference source not found]

The System Design Document (SDD) is written after the initial system decomposition is done, and updated throughout the development. SDD describes the services provided by each subsystem. 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.

SDD is used to define interfaces between teams of developers and serve as a reference when architecture-level decisions need to be revisited. 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.

# Introduction

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).

## Purpose of the System

Aim of the system is to make attendance track simple, quick and trackable by the student as well. It will be simple learn for instructor because it has many options that are used in today’s pen-paper method with extensions of being in digital like getting whole data at a click of finger in the of semesters. Also, it will be quicker than writing every name in the class or checking every student on class. It will be a mobile application, so it will be easier than table using system. In addition, students will be able to track their attendance so that they will be able to know their status about attendance.

## Design Goals

## Definitions, Acronyms, and Abbreviations

AW Sheet: Attendance Watch Sheet

## References

References to existing systems, etc.

# Current Software Architecture

Describe 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.

# Proposed Software Architecture

Documents the system design model of the new system.

## Overview

Present a bird’s-eye view of the software architecture and briefly describes the assignment of functionality to each subsystem.

## System Decomposition

Describe the decomposition into **subsystems and the responsibilities** of each. **This is the main product of system design.**

## Hardware Software Mapping

Describe how subsystems are assigned to hardware and off-the-shelf components. It also lists the issues introduced by multiple nodes and software reuse.

## Persistent Data Management

Describe 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**.

## Access Control and Security

Describe the user model of the system in terms of an access matrix. This section also describes security issues, such as the selection of an authentication mechanism, the use of encryption, and the management of keys.

## Global Software Control

Describe 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.

## Boundary Conditions

Describe 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.)

# Subsystem Services

Describe the **services provided by each subsystem**. 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.

# References

1. “Itslearning App.” *Itslearning - Global*, itslearning.com/us/itslearning-app/.