System design document

Table of contents

Version: 1.0
Date: 2012-03-29
Authors: Oscar Brodefors, Filip Askviken, Rikard Andersson, Emil Nyström

This version overrides all previous versions.

1 Introduction

The system architecture affects the performance, robustness, distributability and maintainability of a system. Therefore, the architecture is important and critical in making a well functioning system (Sommerville, 2007).

1.1 Design goals

The aim for this document regarding the system architecture is to be a helpful tool in the process of implementing the system design. The document is also intended to be useful in the process of introducing a new member in the project.

1.2 Definitions, acronyms and abbreviations

GUI = Graphical user interface

1.3 References

Sommerville, Ian (2007). Software engineering. Eight edition, Pearson Education Limited, Edinburgh Gate, England.

2 Proposed system architecture

In this section we propose a high level architecture.

2.1 Overview

The user of the application is to interact with a GUI on his/her handheld device.

2.2 Software decomposition

2.2.1 General

We aim to have a four-layer structure. With a database and the GUI at opposite ends. The GUI connects to a functional layer which in turn communicates with a database-layer to handle all interaction with the actual database. This to have a more loose connection between the actual database and the methods making the queries.

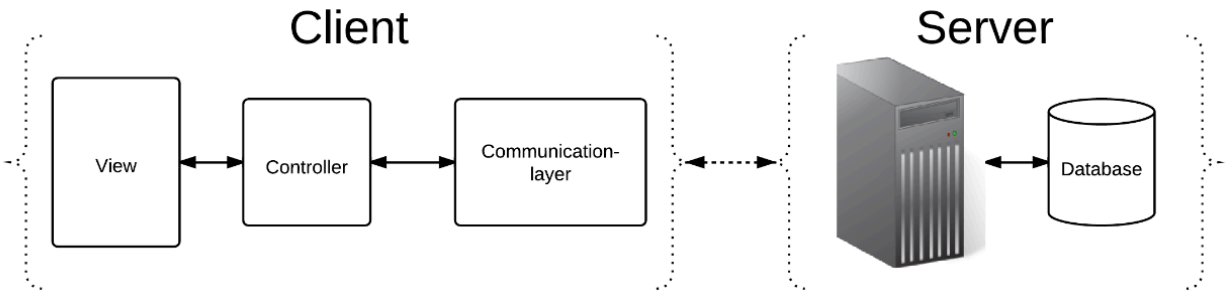
2.2.2 Tiers

The system will be set in two tiers in a server/client-sided manner. The server will only house the database itself while the GUI, functional layer and the databaselayer all reside client-side - the handheld device.

2.2.3 Communication

We aim to have as little communication as possible between the layers. There is no reason for the lower layers to be aware of the layers above itself as all updates/communication will be initiated by the user.

2.2.4 Decomposition into subsystems



2.2.5 Layering

The base of our application is to rely on a database which contains all information regarding the books registered for sale. It will also contain information about users who have books registered for sale, or requested to buy. This will be required to allow the creator of an entry, and the creator alone, to edit and remove said entry.

2.2.6 Dependency analysis

2.3 Concurrency issues

The only person who can actually edit an entry in the database is the creator. Therefore, no concurrency cases will occur.

2.4 Persistent data management

Concerning base functionality we have not identified any data that would require to be persistently managed, as all data regarding the user and the books available are stored server-side.

2.5 Access control and security

Upon first launch of the application a user is required to give up their first and last name which will be registered along with the phone number in the database. Users are later identified by their phone number in every communication with the database.

To modify data (i.e. a book for sale or personal information) in the database you must be the owner of the information. Ownership is ruled by who created the data entry.

To access information about a book for sale no authorisation is needed. All users can access all books for sale.

2.6 Boundary conditions

2.7 References

APPENDIX