PoS Systems

Software Architecture Document

Version 1.2

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 24.11.16 | 1.0 | Creation of document | Sven Baumann |
| 28.11.16 | 1.1 | -minor changes (Architectural Goals and Constraints) | Sven Baumann |
| 29.11.16 | 1.2 | Edit Deployment View | Sandra Kramlich |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Architectural Representation 4

3. Architectural Goals and Constraints 4

4. Use-Case View 4

4.1 Use-Case Realizations 4

5. Logical View 5

5.1 Overview 5

5.2 Architecturally Significant Design Packages 5

6. Process View 5

7. Deployment View 5

8. Implementation View 5

8.1 Overview 5

8.2 Layers 5

9. Data View (optional) 5

10. Size and Performance 5

11. Quality 5

Software Architecture Document

1. Introduction
   1. Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

* 1. Scope

The scope of this SAD is to show the architecture of the PoS application. Affected are the class structure, the use cases and the data representation.

* 1. Definitions, Acronyms, and Abbreviations

tbd to be determined

n/a not avaiable

* 1. References

[MVC Model](https://developer.apple.com/library/mac/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html)

[Classdiagram](https://github.com/PosSystems/pos/blob/master/Classdiagram_MVC.pdf)

[DB Model](https://github.com/PosSystems/pos/blob/master/DBSchema.pdf)

* 1. Overview

This document describes the software architecture to ensure that everybody understands it and new developers can be let into the project easily. It describes how the application is structured, why it is structured like it is and what we do to ensure that our code meets a certain quality standard.

1. Architectural Representation

This project will use the MVC architecture.



[https://developer.apple.com/library/mac/documentation/General/Conceptual/DevPedia- CocoaCore/MVC.html](https://developer.apple.com/library/mac/documentation/General/Conceptual/DevPedia-%09CocoaCore/MVC.html)

1. Architectural Goals and Constraints

The main goal of the MVC architecture is to separate the view from the logic. Therefore the view knows nothing on its own, but gets all information from the logical part. To realize this structure, we’re using Spring.

1. Use-Case View
   1. Use-Case Realizations

n/a

1. Logical View
   1. Overview

[Classdiagram](https://github.com/PosSystems/pos/blob/master/Classdiagram_MVC.pdf)

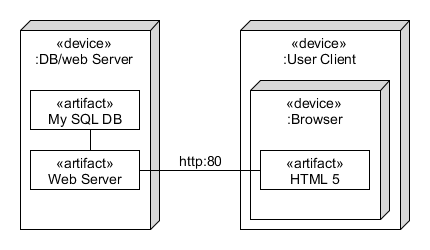
* 1. Architecturally Significant Design Packages

n/a

1. Process View

n/a

1. Deployment View



1. Implementation View
   1. Overview

n/a

* 1. Layers

n/a

1. Data View (optional)

[DB Model](https://github.com/PosSystems/pos/blob/master/DBSchema.pdf)

1. Size and Performance

n/a

1. Quality

n/a