# Software Design Specifications

# for

# Book App

Version 0.1 approved

Prepared by Luca Antonelli

Me, Myself & I

July 5th 2024

Table of Contents

[1. Introduction 1](#__RefHeading___Toc441230972)

[1.1 Purpose 1](#__RefHeading___Toc7234_2709216707)

[1.2 Document Conventions 1](#__RefHeading___Toc441230974)

[1.3 Intended Audience and Reading Suggestions 1](#__RefHeading___Toc441230975)

[1.4 Document Structure 1](#__RefHeading___Toc4280_1579843332)

[1.5 Product Perspective 1](#__RefHeading___Toc441230979)

[2. Static Modeling 2](#__RefHeading___Toc4282_1579843332)

[2.1 Package <Name> with Priority <x/3> 2](#__RefHeading___Toc4284_1579843332)

[2.1.1 <Class Specifier> <Class Name> 2](#__RefHeading___Toc4286_1579843332)

[2.1.2 <Another class Specifier> <Another Class Name> 2](#__RefHeading___Toc4288_1579843332)

[2.1.3 Class Diagram of Package <Package Name> 3](#__RefHeading___Toc4290_1579843332)

[2.2 Package <Name> with Priority <x/3> 3](#__RefHeading___Toc4292_1579843332)

[2.3 Composite Structure Diagram 3](#__RefHeading___Toc4294_1579843332)

[3. Dynamic Modeling 4](#__RefHeading___Toc441230986)

[3.1 Sequence Diagrams 4](#__RefHeading___Toc4296_1579843332)

[3.1.1 Sequence Name 4](#__RefHeading___Toc4298_1579843332)

[3.1.2 Another Sequence Name 4](#__RefHeading___Toc4300_1579843332)

[3.2 StateChart Diagrams 4](#__RefHeading___Toc441230987)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Release Description | Version |
| **Felix Friedrich** | 5.7.24 | Template for Software Engineering Course in ETHZ. | 0.2 |
|  | 5.7.24 | <Write your new release description over here> |  |

# Introduction

## Purpose

*<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SDS, particularly if this SDS describes only part of the system or a single subsystem.>*

## Document Conventions

*<Describe any standards or typographical conventions that were followed when writing this SDS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

## Intended Audience and Reading Suggestions

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SDS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

## Product Perspective

*<Describe the context and origin of the product being specified in this SDS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SDS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

# Static Modeling

## Package <Name>

*<Describe what the package contains and what is it used for. If contents also serve some specific purpose for example: UI boundary, controller, entity, it should be specified here. This text should be no more than a paragraph. Below each class of the package is described separately.>*

### <Class Specifier> <Class Name>



*<Describe why this class is useful and when/how is used by the system.>*

The class attributes are:

* SOME ATTRIBUTE: *<what this is (not just its type, but what is its use)>*
* ANOTHER ATTRIBUTE: ...
* …

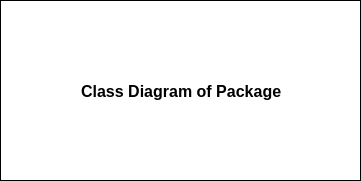
The class operations are:

* SOME OPERATION: *<what this operation does>*
* ANOTHER OPERATION: …
* …

### <Another class Specifier> <Another Class Name>

….

### Class Diagram of Package <Package Name>

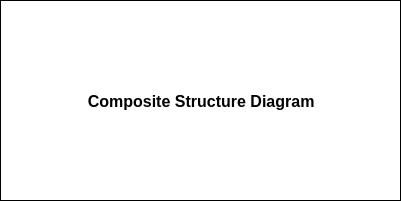


## Package <Name>

**…**

## Composite Structure Diagram

*<Here the composite diagram describing the connections between each package should be provided.>*

**

# Sequence Diagrams

*<Sequence Diagrams are linked to the scenarios and the functional requirements of the SRS document. In this diagrams we are trying to show how the modeling can be used to implement this scenarios and requirements.>*

## Sequence <Name>

*<Describe what this sequence is about.>*

**The functional requirements related to this sequence are:**

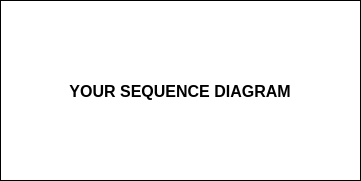
* FREQ-1: FREQ-1 descriptions
* …

**The scenarios which are related to this sequence are:**

* SCN-1: *<Write and describe which scenarios from the previous document relate to this sequence>*
* …

**Scenario Narration**:

*<Narrate your scenario, along with the given diagram below. The narration should be one/two paragraphs long>*

**

## <Another Sequence Name>

**…**

# Interface Modeling

*<Interfaces define how data is exchanged between different parts of the system.>*

## Interface <Name>

*<State what the purpose of this interface is and when it is used. Indicate which modules of the software are involved and which one initiates the communication. Define which protocol is used for communication. Indicate what modes of interaction exist and what types of messages are passed in each mode.>*

**Purpose:** …

**Communication between:** <module X> and <module Y>, initiated by <module X>

**Protocol:** … (e.g. TCP)

**Communication modes:** ...

### <Message Name>

*<State which module is sending the message, which one is receiving it. For each type of message, list the values that are passed, which types they have and if they are required. Define the format of the message, possibly with an example. Also indicate if a response is expected and what type of message that response should be.>*

**Purpose:** …

**Direction:** <module X> to <module Y>

**Content:**

* playerID : integer (required)
* moveType: string (required)
* card: integer

**Format:** as JSON string

**Example:** { "playerID": 2, "moveType": "draw", "card": 8 }

**Expected response:** <other message name>

### *<Another Message Name>*

## *<Another Interface Name>*