Nappy, the ingenious
Software Architecture Document

Revision History

Date	Version	Description	Author
12/11/15	1.0	First version	Mehmet Ali Incekara
28/11/15	1.1	Add SRS ref.	Mehmet Ali Incekara
19/12/15	1.2	Updated all Pictures and Added DB Model	Marc Mahler
19/12/15	1.3	New structure	Mehmet Ali Incekara

Table of Contents

Software Architecture Document

1. Introduction

1.1 Purpose

This document provides an architectural overview of Nappy, the ingenious in aspects of different architectural views.

1.2 Definitions, Acronyms, and Abbreviations

SAD	Software Architecture Document
MVC	Model-View-Controller

1.3 References

GitHub-Docs: https://github.com/nappydevelopment/docs

SRS:
https://github.com/nappydevelopment/docs/blob/master/pdfs/Software%20Requirements%20Specification.pdf

2. Architectural Representation

The project Nappy, the ingenious will use the MVC-principles.

3. Architectural Goals and Constraints

The main goal of this architecture is to separate the view from the logic. The view is “stupid” and knows nothing.
We will use the Framework from JavaFX for our project.

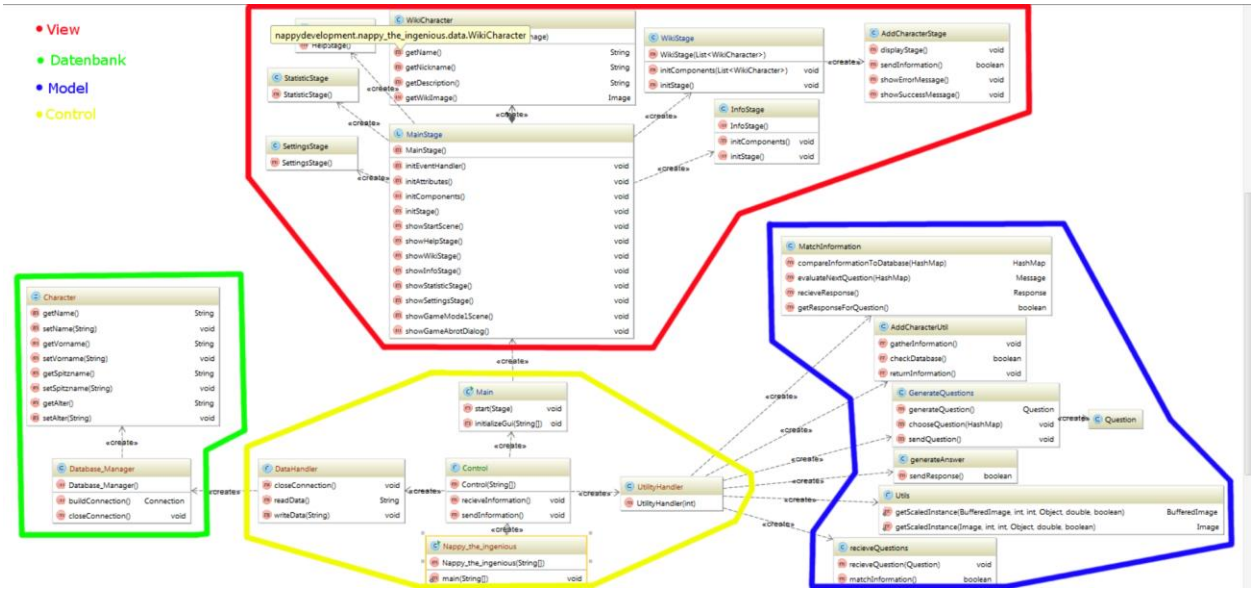
4. Use-Case View

(n/a)

5. Logical View

5.1 Overview

This Class diagram represents how we did plan to design our application.



https://github.com/nappydevelopment/docs/blob/master/pdfs/Class_Diagramm.pdf

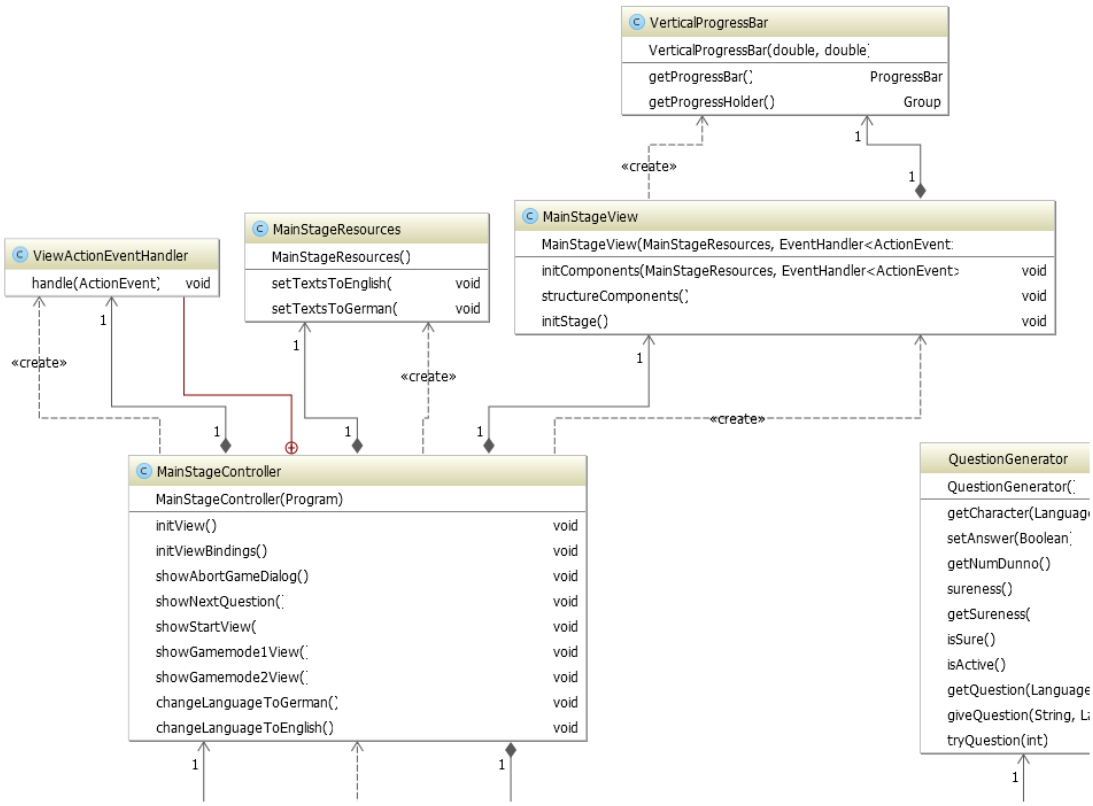
After we updated our project, there was a new class diagram produced. In the new class diagram we completed our work on the MCV principle and used the JavaFX Framework to create it.

The new Class Diagram is pretty huge, so you won't be able to read what is in there in this document. You can follow the provided link to see the full Class Diagram as an SVG and zoom and scroll in it.

https://github.com/nappydevelopment/docs/blob/master/svg/New%20Classdiagram.svg

Our MVC concept consists in many "stages" as they are called in JavaFX. A stage represents one window or one label inside a window. Every Stage has their own MVC concept, which makes it pretty hard to show a detailed view about Model, View, Control and Database. That's why we decided, to add one single stage right here and mention the important parts in the chapters "Separated View" below.

5.2 Separated view



The main stage has his own controller, model and view so every stage is independent. We can easily switch every stage or add new stages.

6. Process View

(n/a)

7. Deployment View

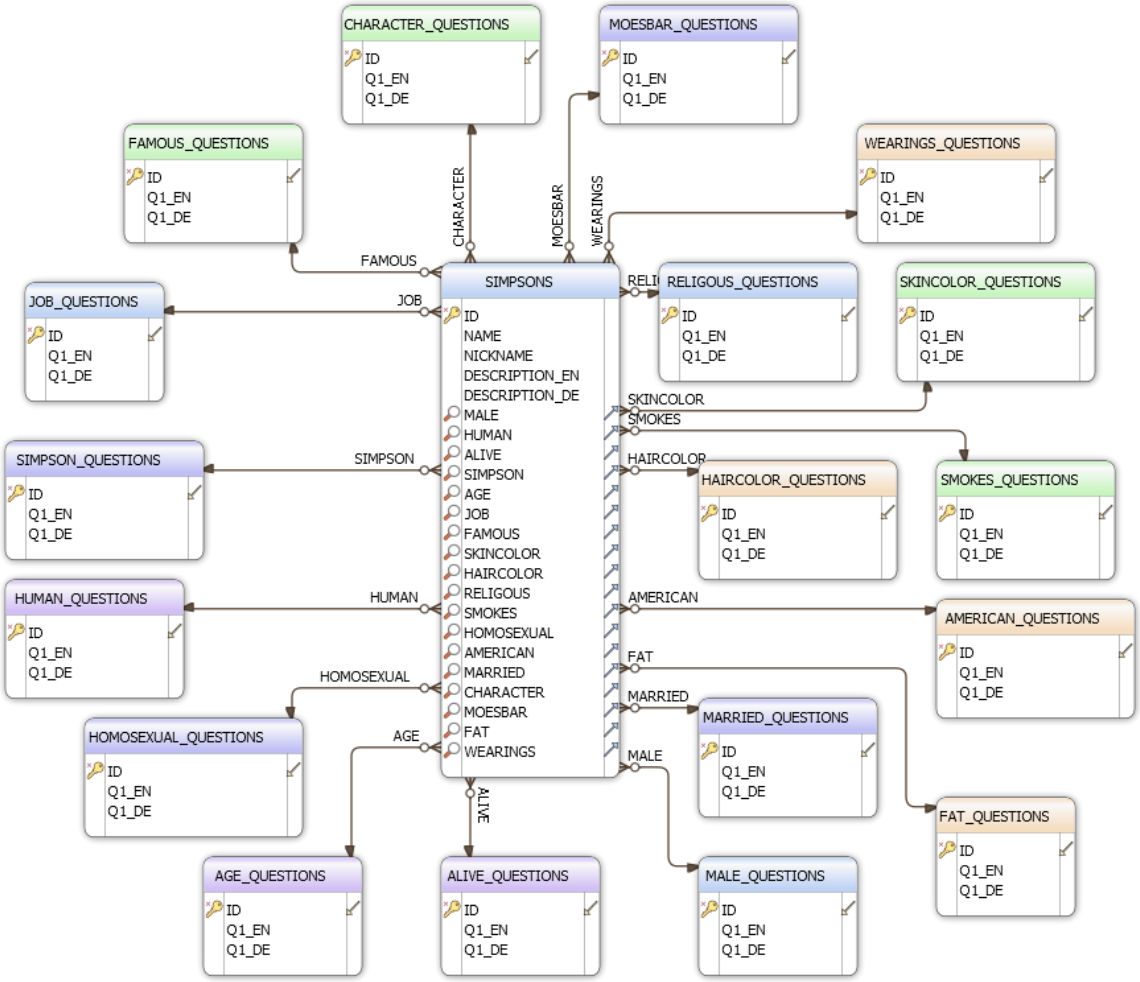
(n/a)

8. Implementation View

(n/a)

9. Data View

For our Database, we also have a model, that might give you a short look at our Database-structure:



https://github.com/nappydevelopment/docs/blob/master/presentation/Database%20Diagram.svg

10. Size and Performance

The size and the performance are really important for our project. We are trying to keep the size of the project as small as possible. Also we will load our database into the memory to increase the performance. So we will need a few seconds until the program starts to load all pictures and the database.

11. Quality

The quality of our project Nappy, the ingenious is also important. We used a lot of sources to get all information about the characters and so our database has a stable foundation.