Software Architecture Document

Fit Up 1.0

Revision History

Date	Version	Description	Author
27.03.17	0.1	Initial version	Yassine Aziani

Table of Contents

1. Introduction

1. Purpose

This document provides an architectural overview of the system behind the Fit Up application.

2. Scope

This document applies to the overall design of the system.

3. References

The template for this document

http://web.cs.wpi.edu/~gpollice/cs4233-a05/rup_sad.html

2. Architectural Representation

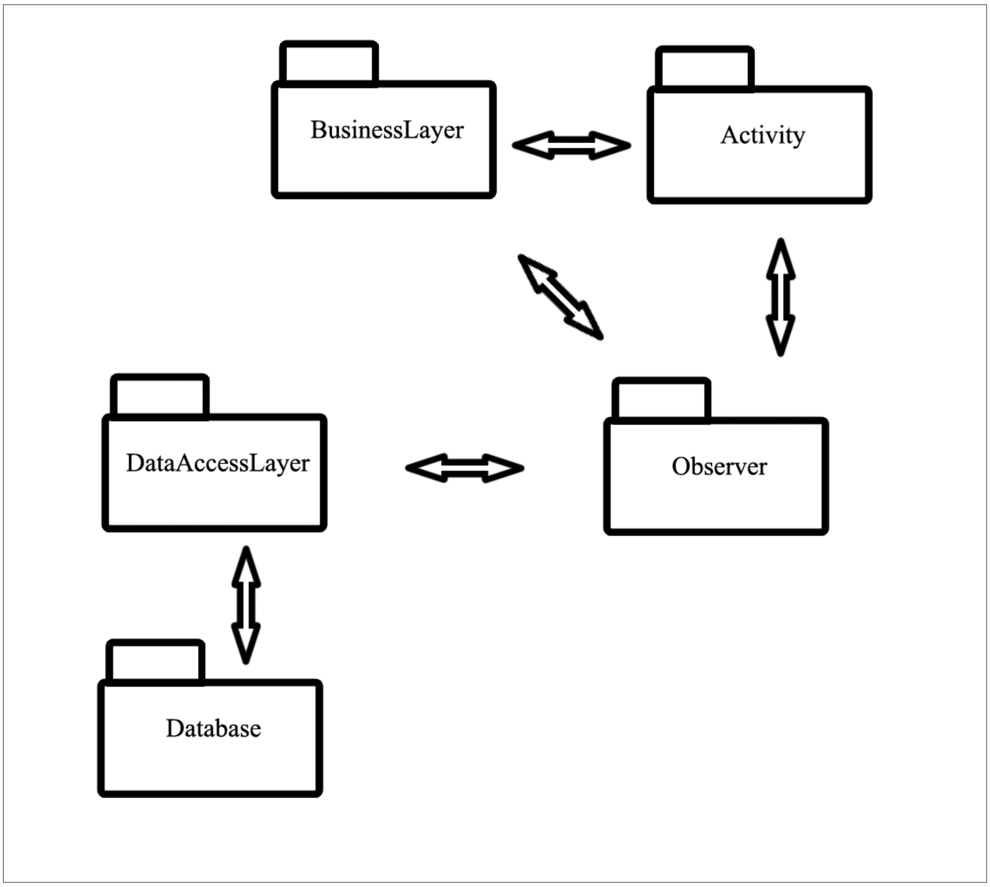
The system was designed using a standard Model-View-Controller pattern, with view containing the visible components, model handling the logic and controller managing the user inputs.

3. Architectural Goals / Constraints

The application was designed with the idea of extensibility in mind.

4. Logical View

1. Overview



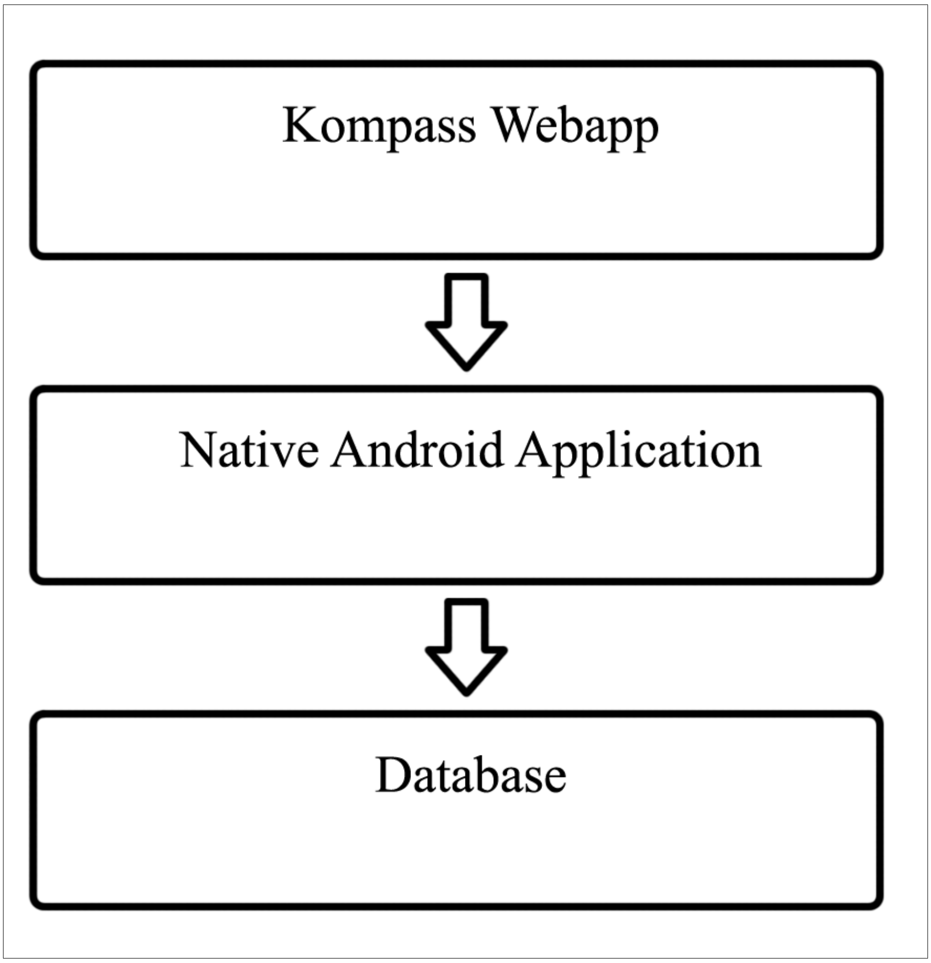
2. Significant Design Packages

The most important packages consist of

1. the BusinessLayer, which contains classes handling the logic behind the functionality,
2. the Activities, containing visual components,
3. Observer, which initializes the execution of the applications functions and
4. the DataAccessLayer, which communicates with the database.

5. Implementation View

1. Overview



2. Layers

1. Kompass Webapp:
   A webapp, which contains the BSA questionnaire and which will be shown on the first registration.
2. Native Android Application:
   The main application, running the motivation methods and handling the user inputs.
3. Database:
   Collected data will be stored within a database, which also provides functionalities for administrators regarding the app's behavior.

6. Size and Performance
   As it is, the application will be used to do scientific research within the TU Darmstadt. It may be extended somewhere in the future.

7. Quality
   The following quality goals have been identified:
   1. Usability:
      Description: the application may not disturb the user and shall be intuitive and easy to use
      Solution: comprehensive usability questionnaires
   2. Stability:
      Description: the application shall not crash or be interrupted
      Solution: automated random input tests (Monkey Test)
   3. Correctness:
      Description: collected data shall be correct and complete
      Solution: JUnit tests to validate functional database operations