

# Architectural Design Document

Ctrl Alt Defeat

June 26, 2023

## 1 Architectural Design Strategy

### 1.1 Approach

We have opted to follow the approach of: Design based on Quality Requirements

### 1.2 Reasons for approach

- This approach allows us to model our system as a solution to an abstract problem (that being the satisfaction of quality requirements). This allows us to make technology-independent design decisions.
- By formulating our design based on quality requirements, we are able to ensure (make a huge step towards ensuring) that the system does indeed meet its quality requirements first and foremost before any code has been written.
- By analyzing the system in terms of quality requirements, it becomes clearer which architectural strategies and patterns (discussed below) are most suitable for the system's implementation.

### 1.3 Reasons to not follow other approaches

- Since our application is dashboard centric, which limits edge case interaction from users, we found it unsuitable to take the approach of designing the system via the generation of test cases. Furthermore, designing test cases would not necessarily help us in identifying appropriate architectural strategies and patterns (at least not to the extent that our favoured approach allows)
- There are relatively few object-like structures that can be decomposed into suitable hierarchical or relationship structures (at least not to enough of an extent) in our system. Hence we felt that a decomposition approach was not necessarily warranted (at least not over our favoured approach, which we deemed to be the most suitable for our purposes)

## 2 Architectural Quality Requirements

### 2.1 heading

Testing out LaTeX

## **3 Architectural Strategies**

We have adopted aspects of several architectural styles, patterns and strategies to help realise our system and its quality requirements.

### **3.1 heading**

Testing out LaTeX

## **4 Architectural Design and Patterns**

### **4.1 heading**

Testing out LaTeX

## **5 Architectural Constraints**

### **5.1 Client-Defined**

- The system needs to consist of at least two separate deployable units.
- The system must be deployed to a private virtual machine with a public IP address (provided by Southern Cross Solutions)
- The system must make use of at least one NoSQL document-based database

### **5.2 Hardware and Operating System Constraints**

- The system is to be hosted on a single virtual machine (running on a single physical server). At a later date this may be expanded to make use of multiple virtual machines running on different physical machines.
- The system must be suitable for and run on an Ubuntu-style Linux operating system

## **6 Technology Choices**

### **6.1 heading**

Testing out LaTeX