Architectural Patterns

Table of Contents

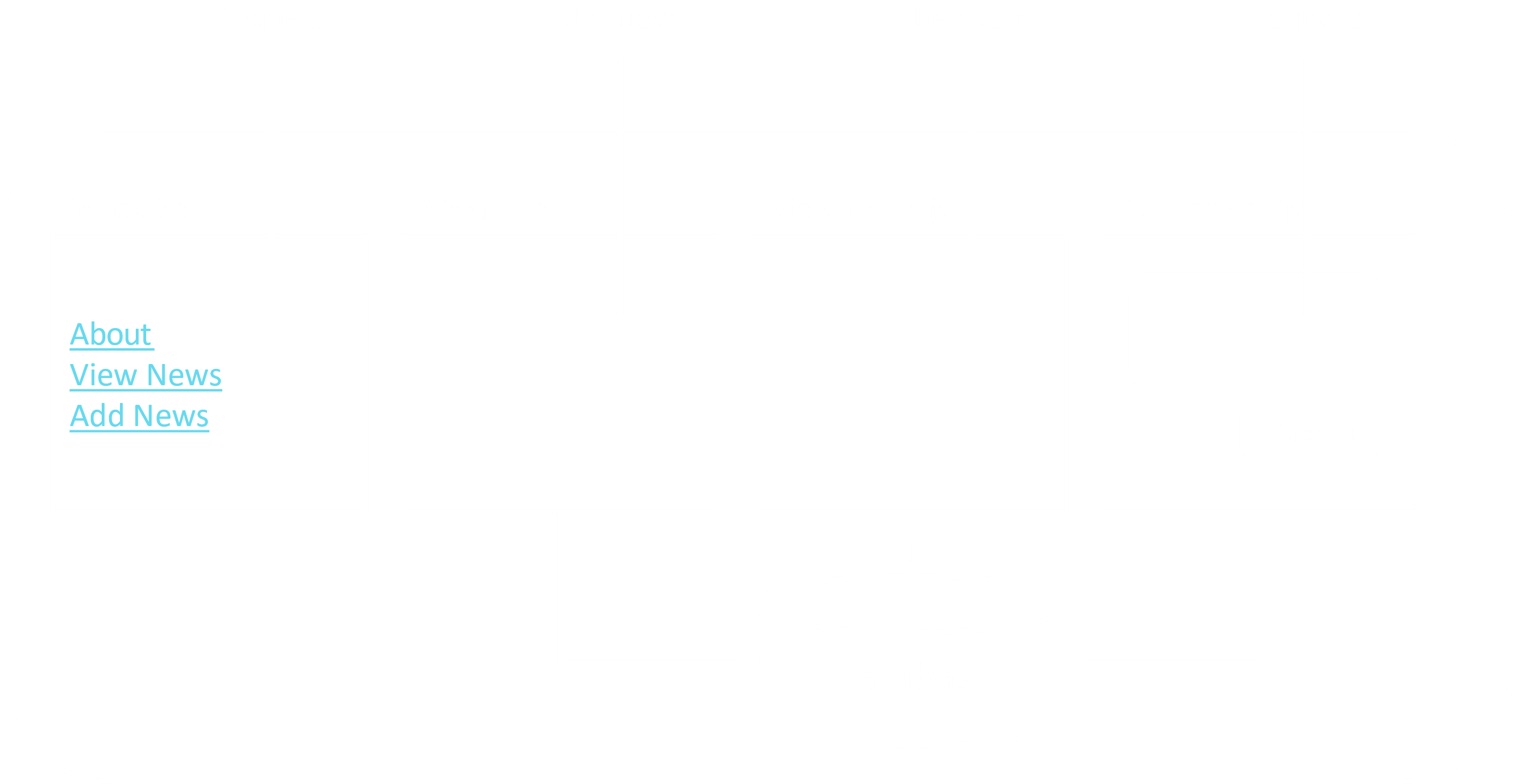
[Model 1 Architecture 2](#_Toc100576773)

[Model 2 Architecture 3](#_Toc100576774)

**Design Patterns** are **templates** that can be used to solve common problems in software design. They are not finished products that can be directly converted to code, but rather methodologies that can be followed to ensure fast and correct development. Specific problems can have multiple solutions, but only some of them are good. Following a design pattern helps remove redundant code, provides proper separation of concerns, allows reusability of code, etc.

## Model 1 Architecture

The **Model 1 Architecture** is a **page-centric architecture**. This means that every page processes its own input. Each page may use a separate JavaBean to do this.



This architecture presents several issues:

* No clear separation of concerns
* Difficult to collaborate
* Difficult to maintain in case of large projects

## Model 2 Architecture

The **Model 2 Architecture** is basically the **MVC Architecture**. As a reminder, the **model** is the database along with code related to reading from and writing to the database, the **view** is used to interface with the user and the **controller** links the other two components and controls them.

Following the Model 2 Architecture ensures clean separation of presentation logic from business logic. There is a **controller servlet** between the client browser and the JSP pages. This servlet dispatches HTTP requests to different **models** based on the request URL, input parameters and application state. The models are responsible for generating the required **view**, which is sent back up the tree.

