Design Documentation

By Sebastian Bokkestijn

Contents

[1.Introduction 2](#_Toc75123618)

[2.Architecture 2](#_Toc75123619)

[2.1.C4 Model 2](#_Toc75123620)

[2.3 SOLID Principles 6](#_Toc75123621)

[2.4 ERD 7](#_Toc75123622)

# 1.Introduction

The project Betty aims to create a web application that anyone practice betting on football matches from real time.

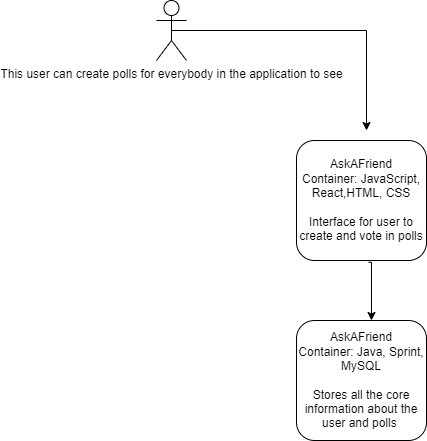
# 2.Architecture

## 2.1.C4 Model

The C4 model was made to show the architecture behind the AskAFriend application. The reasoning for why specific systems and technologies were used are discussed further in the document.

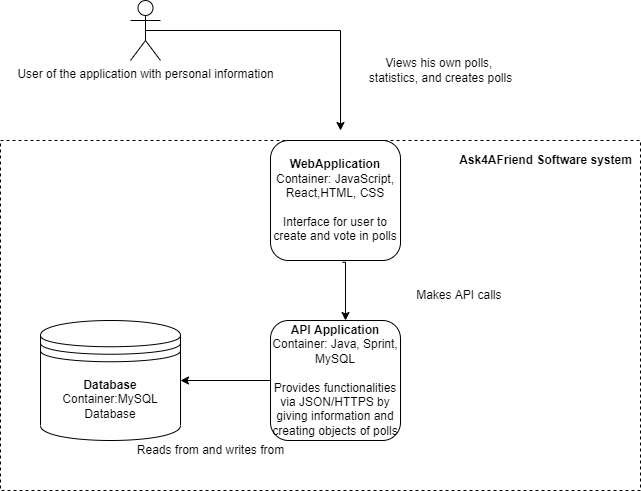
C1: System Context Diagram

The system context diagram shows the software being made, the actors it interacts with the interface of the software and how the front end interacts with the back-end.



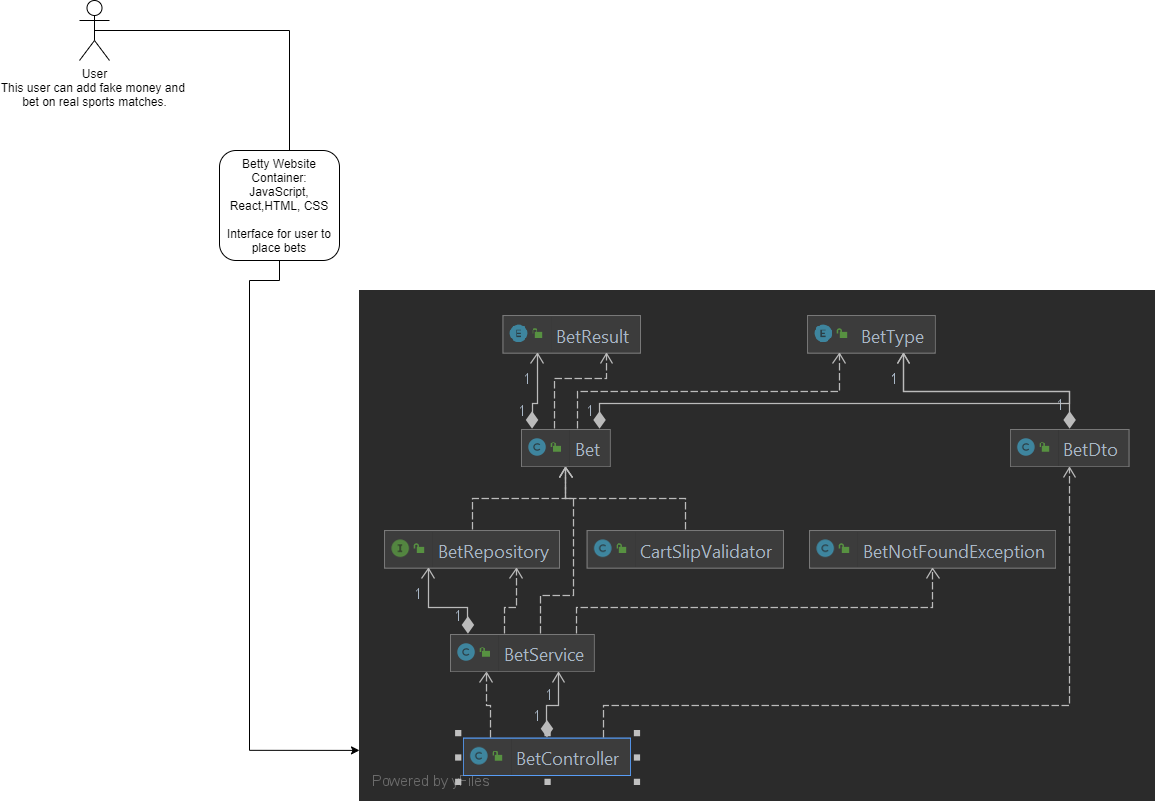
C2: Container diagram

The container diagram provides a deeper look into the Ask4AFriend Web Application software system shown in C1. The application consists of a single page web application made with React, a Spring Boot restful api and a MySQL database to store data.



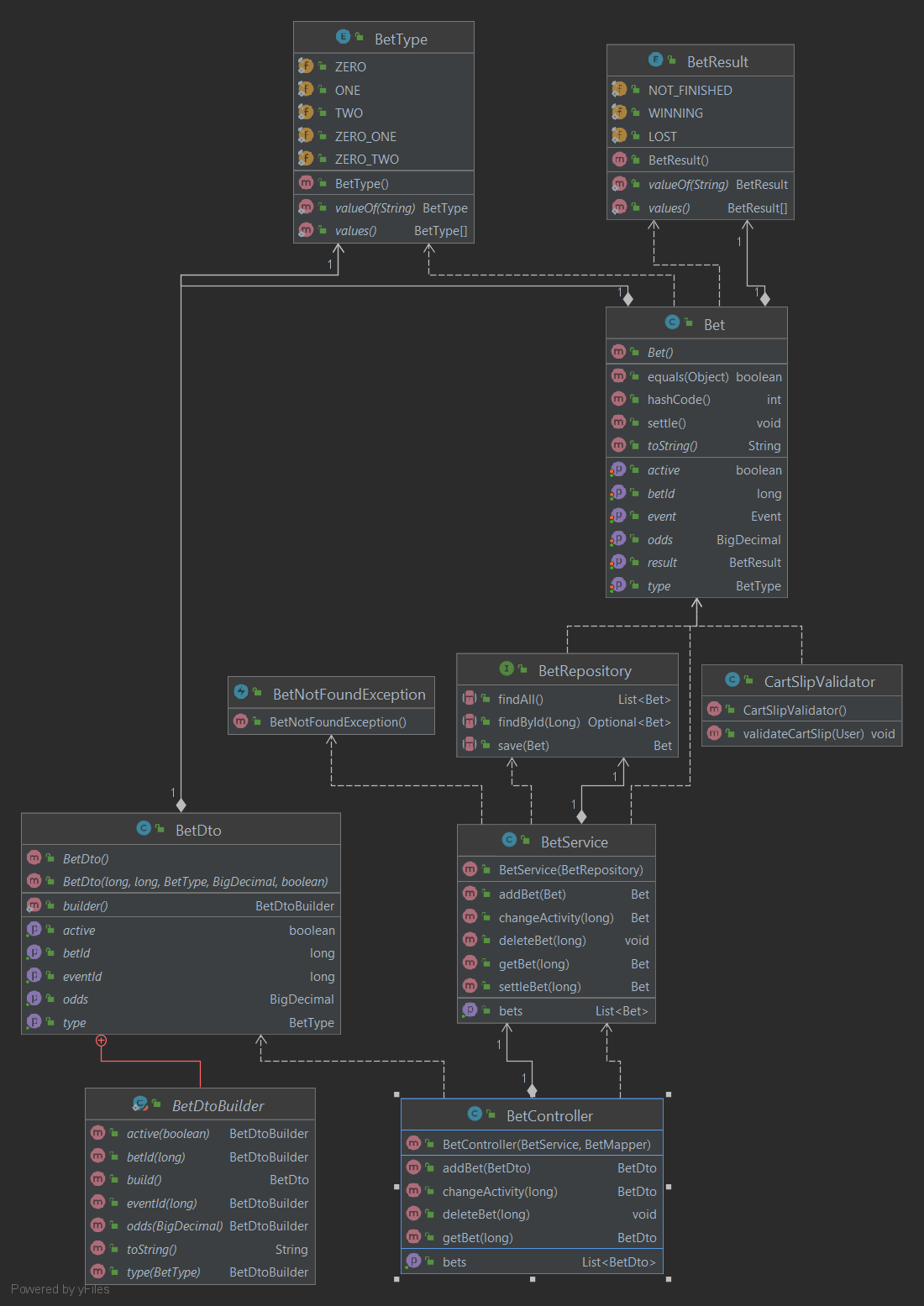
C3: Component diagram

The component diagram goes even deeper and shows the components that make up the API Application container shown in C2. Since there are too many components in the api to fit in this document, the process of making a poll is shown



C4: Code diagram

The C4 diagram is a UML Class diagram. The classes shown are not all of the ones used for the api, but the ones related with the C3 diagram. The relation from Cartslip class to Bet is shown and the bet of interfaces that extend the JpaRepository class created by the spring framework. The use of the service layer is also shown.



## 2.3 SOLID Principles

To ensure solid, a 3 layered design has been implemented. The presentation, logic and data access layer. The logic and data access layers are in the Spring boot application while the presentation layer is solely in thereact application. This avoids any unnecessary, technical coupling between responsibilities and reduces the probability that to change class.es It also lowers the complexity of each change because it reduces the number of dependent classes.

## 2.4 ERD