Java Spring   
Security and Cloud

Baraian Tudor

Anca Giurgiu

Table of contents

Contents

[1. Introduction 2](#_Toc155626724)

[1.1. Project Overview 2](#_Toc155626725)

[1.2. Objectives and Scope 3](#_Toc155626726)

[1.3. Technologies Used 3](#_Toc155626727)

[2. Architecture Overview 3](#_Toc155626728)

[2.1. Description of the Controller-Service-Repository Architecture 3](#_Toc155626729)

[2.2. High-Level Component Diagram (if available) 3](#_Toc155626730)

[3. Security Implementation 3](#_Toc155626731)

[3.1. JWT Authentication Overview 3](#_Toc155626732)

[3.2. Security Configurations 3](#_Toc155626733)

[3.3. Account Management 3](#_Toc155626734)

[4. Front-End Implementation 3](#_Toc155626735)

[4.1. Description of HTML Endpoints 3](#_Toc155626736)

[4.2. Front-End Technologies and Frameworks 3](#_Toc155626737)

[5. Use Cases 3](#_Toc155626738)

[6. Class Diagram 4](#_Toc155626739)

[7. Deployment Diagram 4](#_Toc155626740)

[8. Conclusion 5](#_Toc155626741)

[8.1. Summary of Achievements 5](#_Toc155626742)

[8.2. Final Thoughts 5](#_Toc155626743)

# 

# Project Overview

This project incorporates the functionalities of Spring boot and Spring security.  
Uses html, CSS and JavaScript for the front end part.

It represents a To Do app in which the user can add tasks for a specific day. He/she can choose to add to a list and the tasks will be removed only if the user says they are done with.  
The project showcases the use of authentication token for accessing unsecured endpoints, while being allowed to access only the secured endpoints.  
It showcases the controller-service-repository architecture used in spring boot to interact with the database.  
The project helps users schedule their workload better by offering them a tool for saving and viewing notes for a specific day.

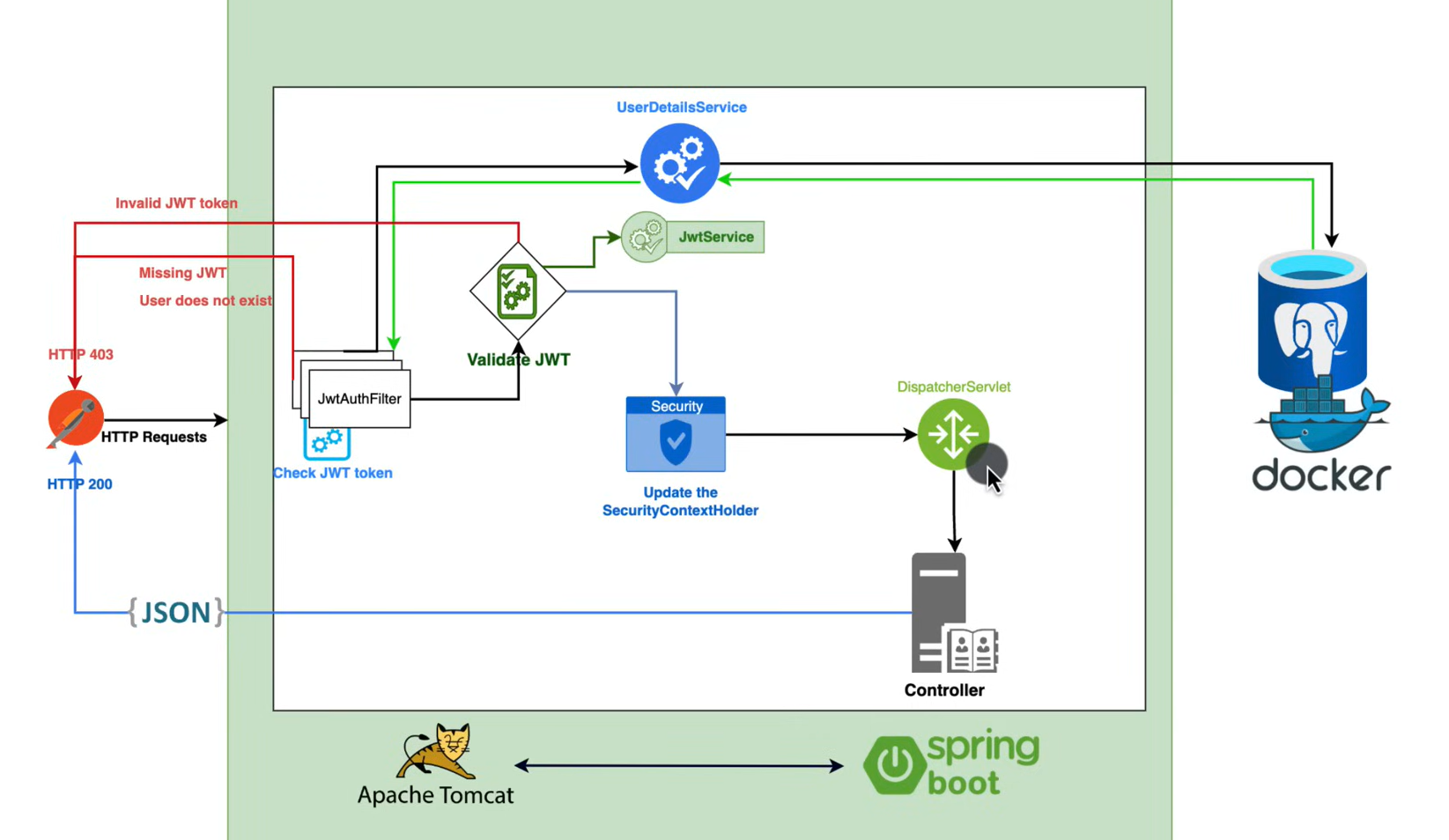
# Architecture Overview

## Description of the Controller-Service-Repository Architecture

## High-Level Component Diagram (if available)

# Security Implementation

## JWT Authentication Overview The main process behind the authentication is to ensure that secured endpoints are accessible and the rest only if they have the authentication token retrieved from the database.



## Security Configurations

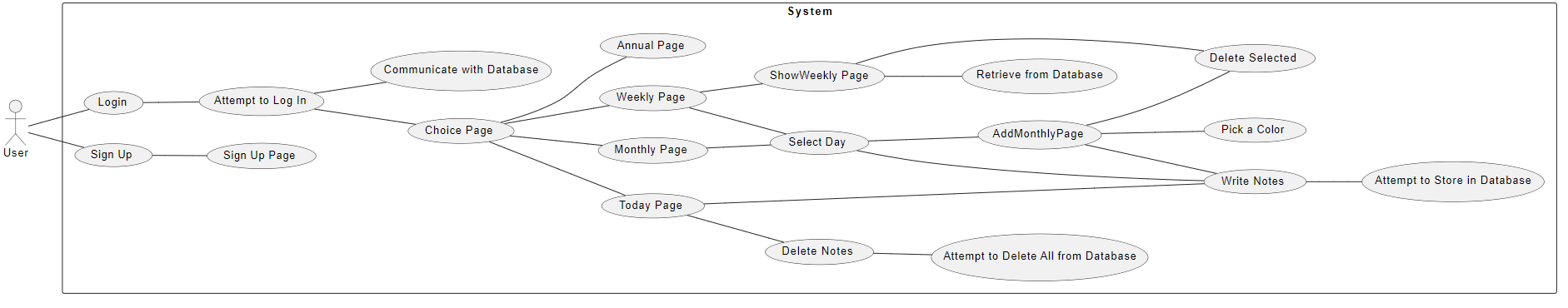
## Account Management

# Front-End Implementation

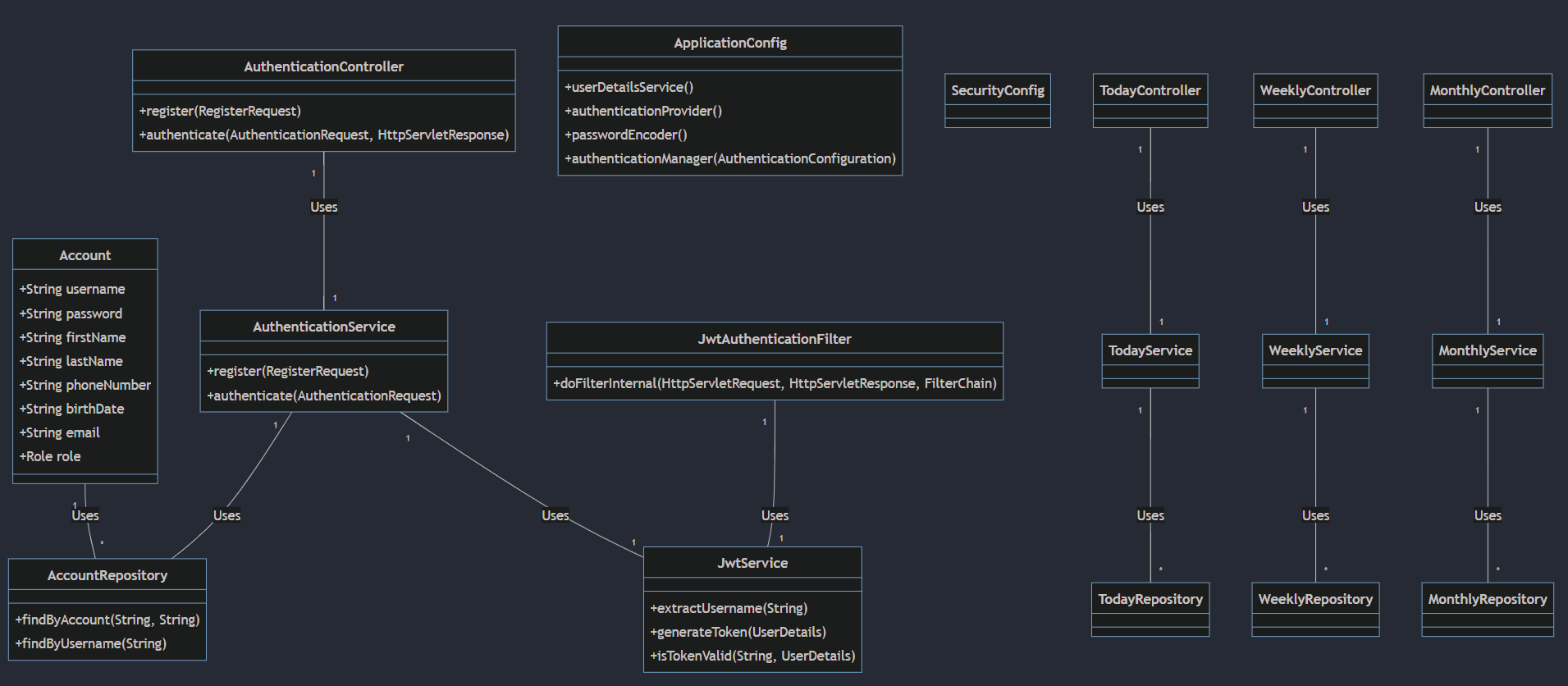
## Description of HTML Endpoints

## Front-End Technologies and Frameworks

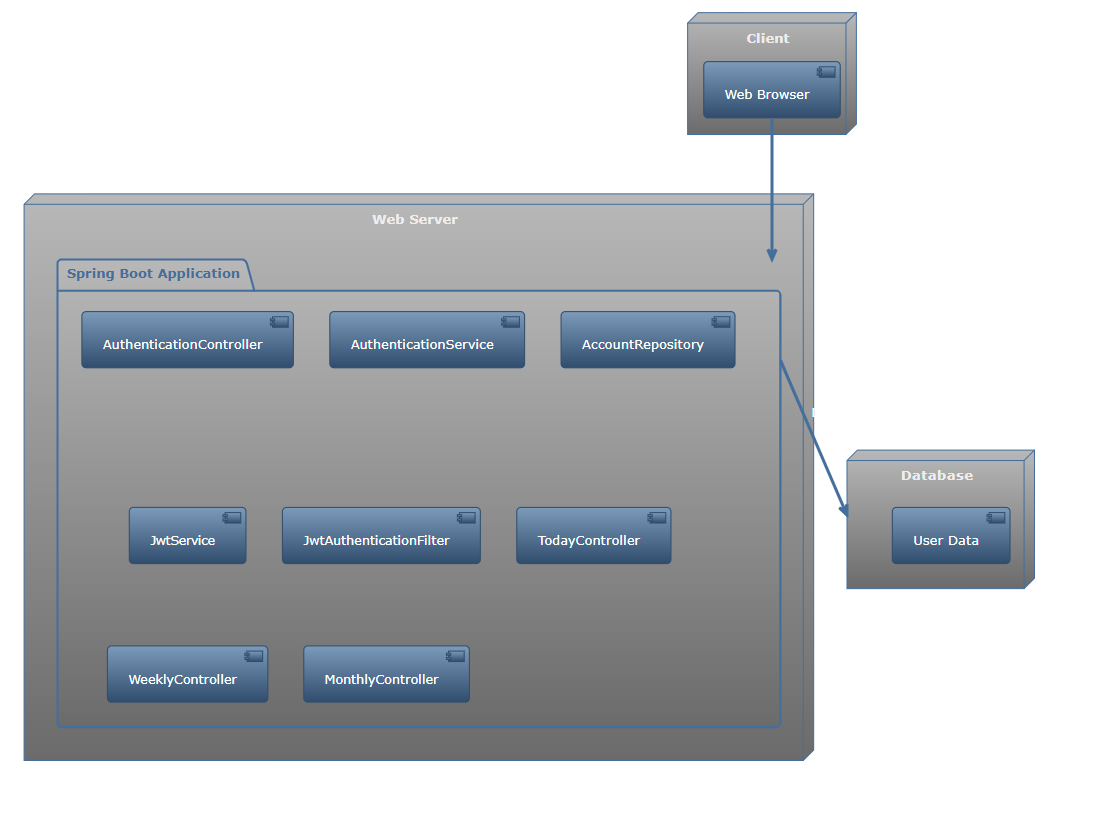
# Use Cases



# Class Diagram



# Deployment Diagram



# Conclusion