

Hund RNG app 2023

20-01-2023

Neopixel

Authored by: Christian Monrad Mortensen



Projektdokument

Indholdsfortegnelse

Projektdokument.....	2
Introduktion	3
Formål	3
Scope.....	3
Målgruppe	3
Definitioner, Akronymmer, Forkortelser osv.....	3
Referencer	4
Overview	4
Resumé	4
Quick Pitch	4
Mål.....	4
Strategier	4
System Overview	4
Kort beskrivelse	4
Nøgle Features	4
Hovedsagelig Bruger Aktiviteter	4
Subsystemer	5
Fremtidige Beredskaber	5
Design Guidelines.....	5
Roller	5
Ansvar	5
System Antagelser	6
Begrænsninger	6
System Afhængigheder	6
Risikoer.....	6
Design Overvejelser	6

Operational Miljø	6
Udviklingsmetoder	6
System Arkitektur og Arkitektoniske Design	7
Diagrammer	7
Hardware Arkitektur	7
Journalhåndtering	7
Design Specifikation	7
Forretningsmæssige Krav	7
Database Design	7
UI / UX Design	8

Introduktion

Denne simple applikation kalder en API som giver et tilfældigt billede tilbage af hund(e).

Formål

Dette er mest en øvelse i visionsstyring og dokumentering af begrundelser for de forskellig valg der er foretaget

Scope

Projektet er ikke ment som mere end simpelheden selv. En enkel side med en knap for at lave et nyt kald og opdater billede på siden til det nyligt kaldte billede.

Målgruppe

Målgruppen er rigtig bredde. Vi sigter efter den generelle danske befolkning, som selvfølgelig også rammer mange andre områder, men som bases skal kunden have en computer med internetadgang og en interesse for at ville se nogle søde hundebilleder.

Definitioner, Akronymmer, Forkortelser osv.

- **API - Application programming interface, tillader et stykke software at kommunikere med et andet software.**
- **GitHub - en kode-hosting-platform til versionskontrol og samarbejde.**

Referencer

1. https://da.wikipedia.org/wiki/Application_programming_interface
2. <https://dog.ceo/dog-api/>
 - a. <https://dog.ceo/api/breeds/image/random> API'en i programmet.

Overview

Resumé

Quick Pitch

Et program som fremviser et hundebillede til bruger med et enkelt tryk.

Mål

Et simpelt program der laver et kald til en API, som så sender en pakke tilbage med et url-link til et hundebillede, som vi fremviser til bruger.

Knap-funktion der erstatter det tidligere hundebillede med et nyt et

Strategier

En enkle programmør burde kunne lave en prototype for sig selv, imens projektledelsen tager sig af det visuelle område. En GitHub repetitorium vil blive oprettet før projektet starter med en side gren til programmørens opgave dele. Forene vil ske når programmøren rapporterer at deres del er færdigt.

System Overview

Kort beskrivelse

Programmet fremviser et billede fra en API og brugen kan ændre det billede med et klik.

Nøgle Features

Et billede på 1000 x 1000 pixel størrelse der kan blive ændret til et andet billede fra API'en med et enkelt tryk fra en knap der er plasteret lige under billedet.

Hovedsagelig Bruger Aktiviteter

Bruger kan kun gøre tre ting

1. Udtrykke deres træthed af deres nuværende billede ved anmodningen af et nyt et.
2. Tag glæde over deres fremviset billede af hund(e).
3. Forlade siden eller start siden op igen.

Subsystemer

Alle nødvendige subsystemer er ikke behøvet at implementere, da de kommer med selve sproget.

Fremtidige Beredskaber

Vores størst bekymring er bare at holde programmet visioner i orden.

Der forventes ingen sikkerhedsrisikoer fra vores side af, da vi ikke behandler nogle personinformationer. Vi regner med sikkerheden der er givet af internet-server-servicesen er gode nok til at søger for at der ingen ændringer kommer til vores program.

Derudover virker det ikke muligt at påvirke API-giverne eller deres database med den vi bruger.

Design Guidelines

Roller

Programmør – udvikler programmet der kan udføre de ønsker vi har.

Designer – opretter og villigholder brugerinterfacene.

Projektleder – opretteholder alt dokumentation, serverforhold, GitHub'en og fremtidige opdateringer.

Ansvar

Server for dog-API'en er ikke vores ansvar, men vi skal holde os opdateret om de laver nogle ændringer.

Internetadgang ansvar ligger hos server-tjenesteudbyder. Vi skal bare holde forholdet positivt mellem os.

System Antagelser

Vi forventer at brugeren benytter en computer med internetadgang og deres maskine har specs til at køre minimumskravene for Google Chrome

Begrænsninger

Behold alt kodning til et minimum og forsøge at holde det til det mindste muligt

System Afhængigheder

En internet-browser (som Explorer, Fire Fox, Chrome, Safari osv.) der kan få adgang til Europa

Risikoeer

Det er muligt at der kan være fejl i områder, som dog-API'en eller internetserver, men der kan vi ikke gøre noget. Og pga. fremtidige opdateringer som vi ikke har indflydelse på, så sætter projektledelsen sig, på den sidste dag af måneden og tjekker efter mulige fejl.

Design Overvejelser**Operational Miljø**

Med hundred procent sikker kan vi sige at programmet kommer til at køre på en server af vores valg.

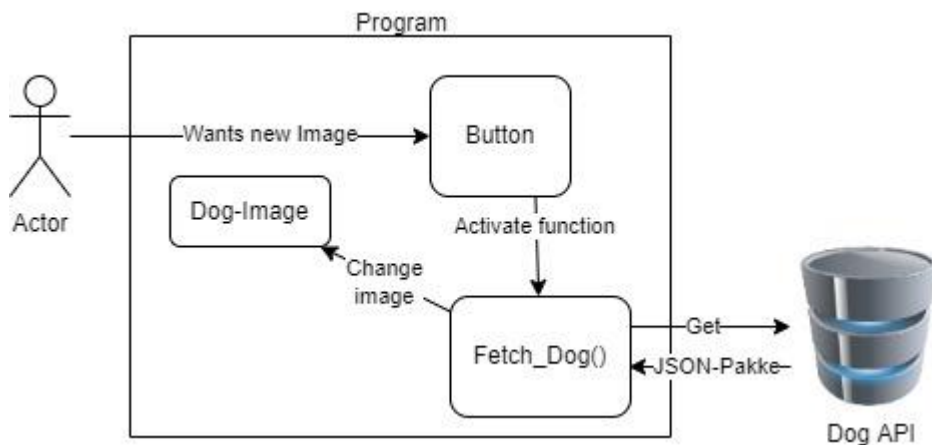
Selve bruger vil brug dette program gennem en browser der kan få forbindelse til vores server. Derfor kan vi ikke påvirke meget, dog er det en fordel at vi ved hvordan de kommer til programmet

Udviklingsmetoder

Udvikling af dette program følger vandfaldsmetoden.

System Arkitektur og Arkitektoniske Design

Diagrammer



Hardware Arkitektur

Vi sjæler ikke noget hardware med dette projekt

Journalhåndtering

Projektlederen har fuldt ansvar for at holde alle dele af Journaler. Dog der er kunne dette dokument når det kommer til Journaler, men de skal også holde øje med alle commits-kommentarer.

Der er ingen planer om at have andre dokumenter til dette projekt

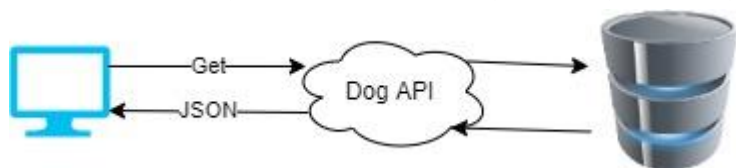
Design Specifikation

Forretningsmæssige Krav

Dette har et non-profit formål, så det er ment som en ultra-lav vedligeholdes, samt kost. Derfor er der sparet i alle områder der er mulige at skære penge fra.

Database Design

Vi har ikke nogen database, dog vi bruge stadigvæk Dog-APIs database.



Så her er det standardformat vi bruger.

UI / UX Design

Som nævnt tidligere, skal designet være meget simpelt, men også læseligt.

