

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matevž Cesar

**Uporabniku prijazno krmiljenje  
pametnih LED trakov**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM  
PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Luka Čehovin Zajc

Ljubljana, 2018

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru naloge izdelajte spletno aplikacijo, ki bo omogočala intuitivno krmiljenje pametnih LED trakov (ki omogočajo naslavljanje posamezne LED diode). Aplikacija naj omogoča urejanje topologije LED traku, ter animiranje grafičnih elementov v istem prostoru. Barva LED diod je nato določena na podlagi barve teh elementov. Za hranjenje in predvajanje animacij predlagajte tudi preprost podatkovni format in razvijte kodo za kodiranje in dekodiranje animacij. Aplikacijo ovrednotite z uporabniško študijo ter interpretirajte njene rezultate.



*Zahvaljujem se mentorju doc. dr. Luku Čehovinu Zajcu za njegovo predanost delu, nesebično pomoč in dragocen čas, ki mi je bil namenjen.*









# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Predstavitev področja . . . . .	2
1.2	Motivacija . . . . .	2
1.3	Cilj dela . . . . .	3
1.4	Pregled strukture . . . . .	3
<b>2</b>	<b>Načrt in gradnja aplikacije</b>	<b>5</b>
2.1	Žični okvirji . . . . .	7
2.2	Uporabljene tehnologije in programska oprema . . . . .	9
2.3	Implementacija v zaledju . . . . .	10
2.4	Implementacija uporabniškega dela . . . . .	12
<b>3</b>	<b>Krmiljenje LED traku</b>	<b>17</b>
3.1	Sestava formata animacije . . . . .	17
3.2	Krmiljenje . . . . .	19
<b>4</b>	<b>Študija uporabniške izkušnje</b>	<b>23</b>
4.1	Naloge uporabniku . . . . .	23
4.2	Anketa . . . . .	24
4.3	Rezultati ankete . . . . .	25
4.4	Intervju . . . . .	29

<b>5 Sklep in nadaljne delo</b>	<b>35</b>
5.1 Sklep . . . . .	35
5.2 Nadaljnje delo . . . . .	35
<b>Literatura</b>	<b>37</b>

# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>LED</b>	light-emitting diode	svetleča dioda
<b>MVC</b>	model-view-controller	model-pogled-krmilnik
<b>SQL</b>	structured query language	strukturirani povpraševalni jezik za delo s podatkovnimi bazami
<b>HTML</b>	hyper text markup language	jezik za označevanje nadbese-dila
<b>CSS</b>	cascading style sheets	kaskadne stilske podloge
<b>JSON</b>	javascript object notation	objektna notacija javascript
<b>IDE</b>	integrated development environment	integrirano razvojno okolje
<b>GUI</b>	graphical user interface	grafični uporabniški vmesnik
<b>DOM</b>	document object model	dokument objekt model
<b>RGB</b>	red green blue	rdeča zelena modra
<b>RLE</b>	run-length encoding	kodiranje s pomočjo zaporedij
<b>SMD</b>	surface-mount device	naprava nameščena na integrirano vezje
<b>TCP</b>	transmission control protocol	protokol za nadzor prenosa
<b>DNS</b>	domain name system	sistem domenskih imen
<b>USB</b>	universal serial bus	univerzalno serijsko vodilo



# Povzetek

**Naslov:** Uporabniku prijazno krmiljenje pametnih LED trakov

**Avtor:** Matevž Cesar

Pametni LED trakovi ponujajo možnost upravljanja vsake posamezne diode in tako omogočajo ustvarjanje najrazličnejših barvnih animacij in vzorcev. Njihovo krmiljenje je omejeno na nizkonivojsko programiranje posameznih diod ali na uporabo namenske programske in strojne opreme, ki je namenjena profesionalni rabi. V okviru diplomske naloge smo razvili spletno aplikacijo, ki uporabnikom pomaga ustvarjati enostavne animacije za LED trakove preko enostavnega uporabniškega vmesnika. Rezultati študije uporabniške izkušnje kažejo, da je bila razvita aplikacija uporabnikom zanimiva in preprosta za uporabo.

**Ključne besede:** LED trak, spletna aplikacija.



# Abstract

**Title:** User-friendly control of smart LED strips

**Author:** Matevž Cesar

Smart LED strips offer the ability to control each individual diode and can be used to create many different color animations and patterns. The controlling of these strips is currently limited to low-level programming of individual diodes or proprietary software and hardware, which is meant for professional use. Our goal was to create a web application that would allow users to create simple animations for smart LED strips, by using a simple user interface. The results of our usability study show that users found our application useful and simple to use.

**Keywords:** LED strip, web application.





# Poglavje 1

## Uvod

LED trakovi so na trgu že dolgo časa, zadnje čase pa postajajo vse bolj zaželeni, saj je dostopnost vedno večja, cena pa vedno nižja. Kupiti je mogoče mnogo različnih dolžin trakov po relativno nizki ceni. Gre za gibljivo vezje z nameščenimi svetlečimi diodami, na zadnji strani pa ima ponavadi lepilni trak za lažje nameščanje. Trakovi so namenjeni mnogim različnim načinom uporabe, ki pa so tudi odvisni od tipa posameznega traku. Enobarvni, nenaslovljivi trakovi lahko svetijo samo z eno barvo in so primerni za stalno osvetljavo prostora ali predmeta, na primer ohišja računalnika. Večbarvni, nenaslovljivi trakovi lahko prikazujejo RGB barve in jih dinamično spreminjajo. Taki tipi trakov imajo ponavadi priložen daljinec, s katerim lahko določimo barvo traku, svetilnost ter nekaj vnaprej določenih efektov, npr. utripanje. Uporabljajo se predvsem, ko želimo v prostoru ustvariti določen ambient. Večbarvni, naslovljivi trakovi omogočajo upravljanje vsake posamezne diode. S takimi trakovi je mogoče ustvariti poljubne animacije [11] in se zato lahko uporabljajo za najrazličnejše namene. Ustvarjanje ambientov z LED trakovi lahko po nekaterih študijah vpliva na človekovo počutje in celo pripomore k boljšemu spancu [6]. Taki trakovi so bili preizkušeni tudi v avtomobilski industriji, in sicer za signaliziranje povratnih informacij o hitrosti [9]. Še ena zanimiva uporaba je interaktivno krmiljenje traku glede na določene dogodke v računalniški igri. Uporabniki so računalnik dojemali

kot bolj človeškega, ko je na dogodke računalniške igre reagiral z določenimi barvami [10]. Več takih trakov skupaj lahko uporabimo tudi za izdelovanje svetlobnih predstav, oglaševanje, osvetljevanje odrov, ustvarjanje inteligentnih ambientov, itd. Nadzorovanje in upravljanje takih trakov pa se izvaja zelo nizkonivojsko. Potrebna je povezava z mikrokrmilnikom, ki krmili posamezno diodo na traku. Če hočemo z njimi upravljati na višjem nivoju, potrebujemo namensko programsko opremo.

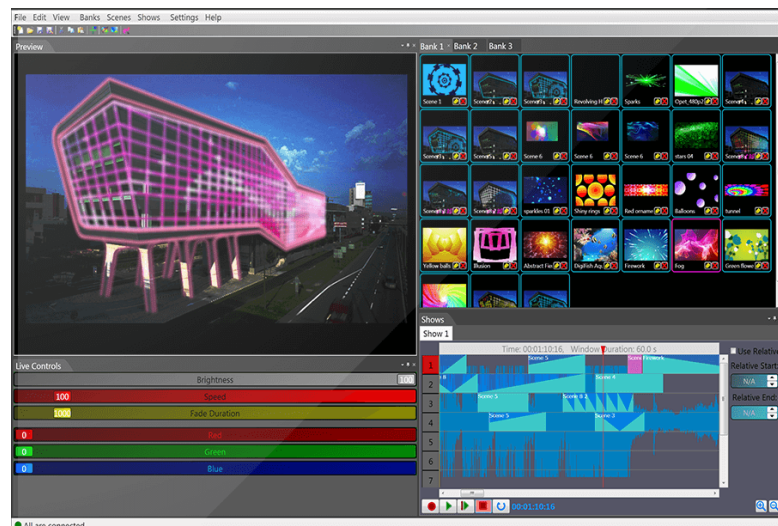
## 1.1 Predstavitev področja

Za kreiranje poljubnih animacij obstaja programska oprema z imenom **LED strip studio** [8]. Programska oprema se prodaja skupaj z namenskimi krmilniki, ki podpirajo upravljanje s 40.000 diodami. Aplikacija ima mnogo funkcionalnosti za animiranje barvnih efektov in podpira prikazovanje besedil in videoposnetkov na platna, ki so sestavljena iz posameznih LED trakov. Možna je tudi sinhronizacija animacij z glasbo in postavljanje trakov v tri-dimenzionalni prostor za lažje animiranje (Slika 1.1). Programska oprema je aplikacija za Windows, ki je namenjena predvsem profesionalni uporabi. Uporablja se za osvetljava večjih objektov in animiranje odrov na različnih prireditvah. Ena zadnjih večjih prireditev, kjer so uporabljali to opremo, je bila Evrovizija 2018.

## 1.2 Motivacija

Za ustvarjanje določene animacije je potrebno nizkonivojsko krmiljenje vsake diode na LED traku, ki pa je zamudno in zahteva znanje programiranja. Razlog za razvoj aplikacije je priskrbeti odprtokodno in zastonjsko rešitev, ki bi omogočala izdelovanje animacij za LED trakove na višjem nivoju v obliki spletne aplikacije. Aplikacija bi uporabnikom z manjšim znanjem o programiranju omogočila ustvarjanje animacij po meri za lastne LED trakove. Ciljni uporabniki so torej hobi uporabniki in interaktivni umetniki, ki uporabljajo

LED trakove za pametno osvetljavo ali v okviru multimedijskih projektov.



Slika 1.1: Zaslonski posnetek vmesnika aplikacije LED Strip Studio

### 1.3 Cilj dela

Cilj dela je ustvariti spletno aplikacijo, ki bi omogočala ustvarjanje animacij za poljubno dolg LED trak. Uporabniku bi tudi omogočala, da določi topologijo traku, in na njem ustvari animacijo. Podpira naj tudi shranjevanje in izvoz animacije v preprost format, ki ga bo uporabil krmilnik za prikaz animacije. Zanimala nas bo tudi uporabniška izkušnja aplikacije, zato bomo izvedli študijo uporabniške izkušnje, s katero bomo pridobili povratne informacije o uporabnosti in intuitivnosti aplikacije ter o možnihboljšavah.

### 1.4 Pregled strukture

V sledečem poglavju smo predstavili načrtovanje programskega dela aplikacije, uporabljene tehnologije, nato pa še njeno izvedbo v zalednem in uporabniškem delu. Tretje poglavje je namenjeno predstavitvi mikrokrmilnika in

LED traku, ki smo ju uporabili. V četrtem poglavju najprej definiramo potek uporabniške študije in nato analiziramo njene rezultate. Zadnje poglavje vsebuje sklepe ugotovitve diplomske naloge in ideje za nadaljne delo.

## Poglavje 2

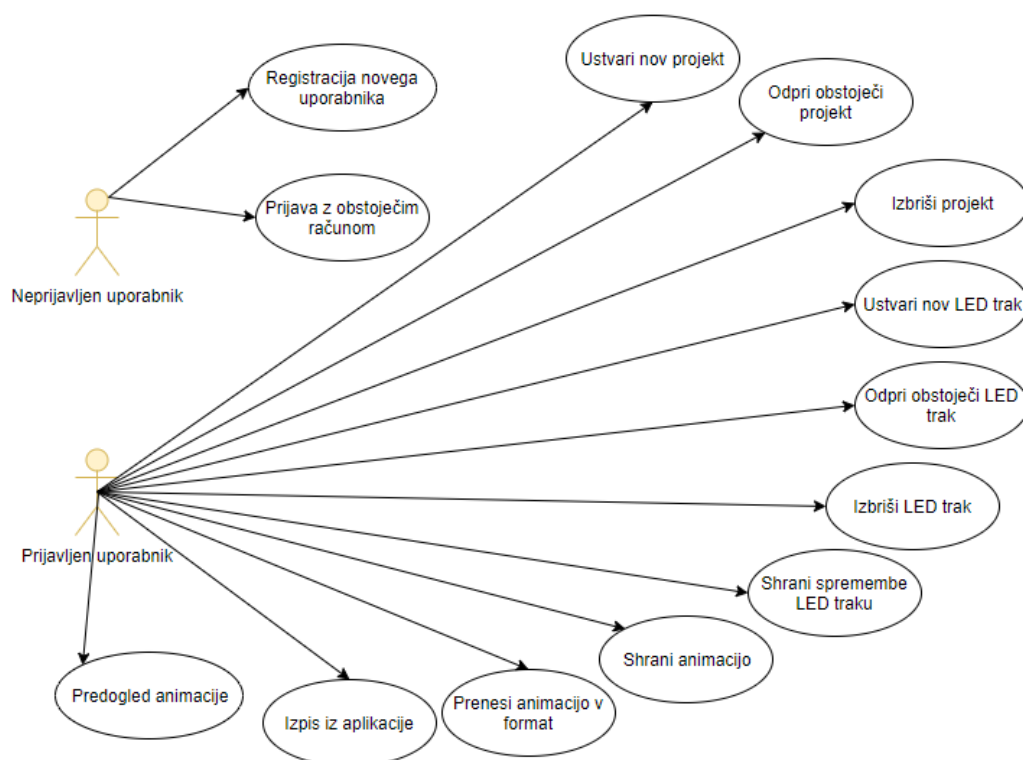
# Načrt in gradnja aplikacije

Da bi dosegli boljšo intuitivnost in enostavnost aplikacije, smo se odločili, da ločimo določanje topologije traku in ustvarjanje animacije na dva pogleda. V prvem pogledu naj bo torej možno prosto določanje topologije LED traku, v drugem pa trak ostane v določeni poziciji, premikati pa je mogoče le objekte, s katerimi animiramo. Na podlagi tega, smo zastavili nekaj funkcionalnosti, ki naj jih aplikacija vsebuje, da bo predstavljala neko smiselno zaključeno celoto.

- **Postavitev LED trakov v prostor** - uporabnik lahko ustvari virtualni LED trak s poljubnim številom diod in določi njegovo topologijo z interaktivnim premikanjem traku.
- **Ustvarjanje animacije** - uporabnik lahko ustvari animacijo, ki se izvede na trenutno izbranem traku. Animira naj se osnovne oblike, ki se jim določi barvo, pozicijo, velikost in rotacijo. Ko oblike na platnu prečkajo LED trak, se na traku prižge dioda v barvi, ki ustreza presečni obliki.
- **Predogled animacije** - možen je predogled končne animacije, ki bolj nazorno prikaže izgled animacije na fizičnem traku.
- **Izvoz animacije** - animacijo je možno izvoziti v format, ki ga bo prebral krmilnik LED traku.

- **Registracija in prijava uporabnika** - želimo, da se delo uporabnika med različnimi sejami ohrani, zato se v ta namen uvede registracija in prijava uporabnika v sistem. Na ta način zagotovimo shranjevanje dela za specifičnega uporabnika.
- **Upravljanje projektov** - naša želja je, da uporabnik lahko dela na več animacijah hkrati, zato omogočimo ustvarjanje več projektov. Projekti so neodvisni drug od drugega, in vsebujejo podatke o določeni animaciji. Za določen projekt je možno ustvariti več različnih dolžin in oblik trakov.

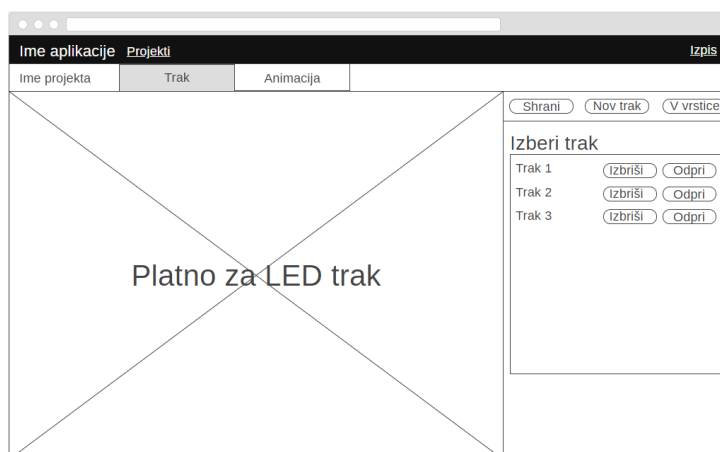
Za lažjo vizualizacijo, smo iz določenih funkcionalnosti zastavili še diagram primerov uporabe (Slika 2.1) za neprijavljenega in prijavljenega uporabnika.



Slika 2.1: Diagram primerov uporabe

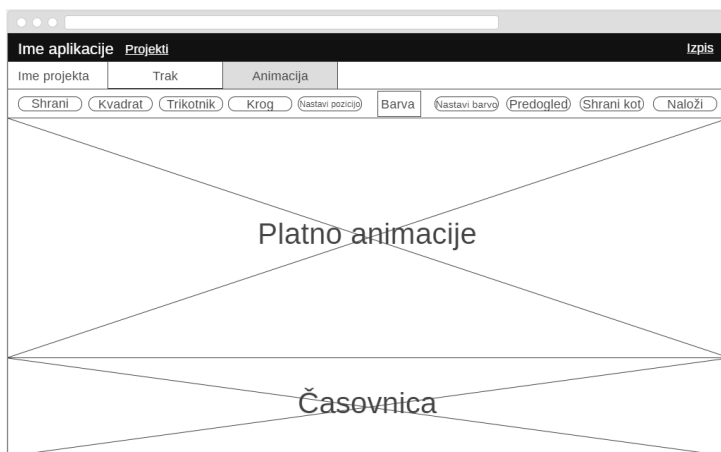
## 2.1 Žični okvirji

Žične okvirje smo določili za glavne poglede naše aplikacije. Najprej smo definirali videz projektne strani. Projektna stran vsebuje dva zavihka. Prvi je namenjen ustvarjanju traku in njegovi postavitvi v dvodimenzionalnem prostoru, drugi pa ustvarjanju animacije nad izbranim trakom. Na Sliki 2.2 je prikazan žični okvir prvega zavihka, ki je namenjen ustvarjanju in določanju topologije posameznega LED traku. Večji del leve strani tega zavihka je namenjen platnu po katerem bo uporabnik z miško premikal diode traku v poljubno obliko. Na desni strani je na vrhu orodna vrstica, ki ima tri gumbe. Prvi gumb na levi shrani trenutno obliko odprtega traku. Drugi je namenjen ustvarjanju novega traku. Ob pritisku nanj se prikaže pojavno okno, v katerem uporabnik določi ime traku in število diod, ki naj jih vsebuje. Tretji gumb je namenjen samodejnemu postavljanju traku v določeno število vrstic. Uporabnik lahko s klikom določi število vrstic v pojavnem oknu, ki jih želi predstaviti s trakom. Trak se ob potrditvi postavi v ustrezno obliko. Pod orodno vrstico, se nahaja seznam vseh kreiranih trakov na trenutnem projektu. Vsak trak v seznamu ima poleg imena dva gumba. Prvi gumb izbriše trak iz projekta, drugi pa naloži trak v platno.



Slika 2.2: Žični okvir projektne strani - zavihek postavitve LED traku

Na Sliki 2.3 je prikazan žični okvir drugega zavihka. Zavihek ima na vrhu orodno vrstico. Prvi gumb v orodni vrstici shrani trenutno stanje animacije. Sledijo mu trije gumbi, ki na platno in časovnico dodajo osnovno obliko, kvadrat, krog ali trikotnik. S klikom na naslednji gumb nastavimo trenutne lastnosti izbrane oblike, za trenutno izbrani čas na časovnici. Sledi polje za izbiro barve in gumb, ki ob kliku pobarva izbrano obliko z barvo, ki je določena v barvnem polju. V nadaljevanju vidimo gumb, ki sproži predogled animacije. Ob predogledu se ustvarjena animacija prikaže na diodah virtualnega traku. Na koncu sta še dva gumba, ki upravljata končno animacijo. Prvi gumb sproži prenos animacije v datoteko, katere vsebino nato preberemo z krmilnikom, drugi gumb pa pošlje podatke animacije neposredno na krmilnik, in tako sproži takojšnje predvajanje animacije na fizičnem traku. Pod orodno vrstico je postavljeno platno, na katerem se izrisujejo oblike, ki jih animiramo, in trenutno izbrani trak. S klikom na posamezno obliko se nam prikažejo nadzori za premik oblike, njeno velikost in rotacijo. Na dnu je časovnica, ki prikazuje lastnosti posamezne oblike glede na čas.

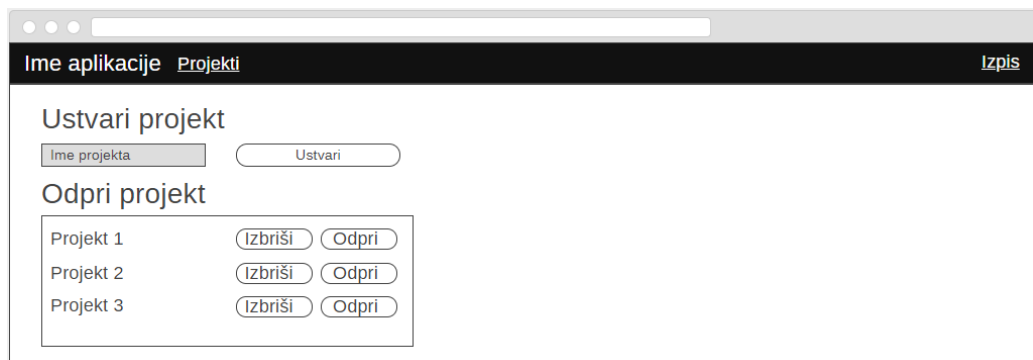


Slika 2.3: Žični okvir projektne strani - zavihek animacije

Določili smo še stran, ki omogoča ustvarjanje in upravljanje s projekti (Slika 2.4). Na vrhu je vnosno polje za poimenovanje novega projekta, poleg njega pa gumb, s katerim ustvarimo in odpremo nov projekt. Spodaj



se nahaja seznam vseh ustvarjenih projektov uporabnika. Vsak projekt v seznamu ima zraven imena še dva gumba. Prvi gumb projekt izbriše, drugi pa ga odpre.



Slika 2.4: Žični okvir strani projektov uporabnika

## 2.2 Uporabljene tehnologije in programska oprema

1. **ASP.NET MVC** je Microsoftovo odprtokodno ogrodje za razvoj spletnih aplikacij, ki je napisano v jeziku C#. Primarno uporablja gonilnik za predloge Razor, s katerim je sprednji del aplikacije mogoče razbiti na več delov in jih po potrebi tudi sestavljati. Tako se izognemo ponavljajoči kodi. Ogrodje uporablja programsko arhitekturno obliko MVC (*model-view-controller*). Aplikacije, ki uporabljajo to obliko so sestavljene iz treh glavnih delov:
  - *Model* - predstavlja strukturo podatkov, ki jih napolni *Controller*, uporabi pa jih *View*.
  - *View* - prikaže podatke modela v nekem vmesniku in omogoča manipulacijo teh podatkov.
  - *Controller* - se odziva na manipulacijo s podatki, nad njimi izvaja logiko in napolni podatke modela.

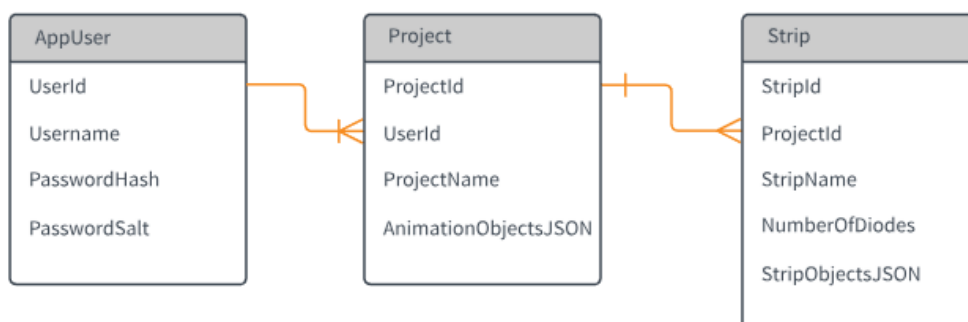
2. **SQL Server Express** je zastopna verzija Microsoftovega sistema za upravljanje relacijske podatkovne baze SQL Server. Sistem izvaja poizvedbe nad podatkovno bazo in uporablja jezik Transact-SQL, ki je razširitev jezika SQL.
3. **HTML** je označevalni jezik za ustvarjanje spletnih strani in aplikacij. Spletne strani so sestavljene iz HTML elementov, s katerimi definiramo njihovo strukturo in vsebino, te elemente interpretira brskalnik in jih zatem prikaže uporabniku. Videz teh elementov ponavadi prilagodimo z jezikom CSS. Za dinamično nadzorovanje vsebine na strani pa se pogosto vključi tudi jezik Javascript. Najnovejša različica jezika je HTML 5, ki vsebuje podporo za multimedijske in grafične vsebine.
4. **Bootstrap** je odprtokodna knjižnica, ki je namenjena lažjemu oblikovanju spletnih strani. Vsebuje predloge za pogosto uporabljene elemente, ki so napisane v jezikih HTML in CSS.
5. **Javascript** je dinamični, šibko tipizirani programski jezik, ki se izvaja v brskalniku in se uporablja za dinamično spreminjanje vsebin na spletni strani. Največkrat uporabljena knjižnica, ki jo uporabljamo tudi mi, je jQuery. Ta knjižnica poenostavi upravljanje z elementi DOM in dogodki, ter upravljanje z Ajaxom, ki omogoča asinhrono nalaganje novih vsebin. Poleg te knjižnice bomo uporabili še knjižnici **Fabric.js** [5] in **Timeline.js** [12]. Fabric.js je knjižnica za enostavno izrisovanje in manipulacijo objektov na HTML 5 platnu, Timeline.js pa je knjižnica, ki vsebuje uporabniški vmesnik za časovni trak in omogoča lažje kreiranje animacij.

## 2.3 Implementacija v zaledju

Aplikacija je zasnovana tako, da večinoma teče v brskalniku. Zaledni del aplikacije skrbi za usmerjanje (ang. routing) aplikacije in shranjevanje podatkov v bazo. Shranjevati je treba podatke o uporabnikih, ustvarjenih projektih in

podatke o trakovih in animacijah, ki so shranjene na posameznem projektu. Aplikacijo smo zasnovali tako, da ima lahko en uporabnik več projektov, en projekt pa več trakov. Prišli smo do enostavnega podatkovnega modela (Slika 2.5), ki je sestavljen iz treh tabel:

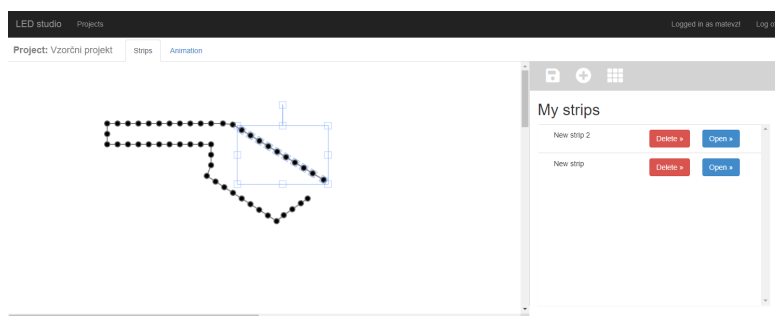
- **AppUser** tabela hrani uporabniško ime *Username*, zgoščeno vrednost gesla *PasswordHash*, sol gesla *PasswordSalt* in enolično oznako *UserId*.
- **Project** tabela hrani ime projekta *ProjectName*, objekte animacije v formatu JSON *AnimationObjectsJSON*, referenco na uporabnika *UserId* in enolično oznako *ProjectId*.
- **Strip** tabela hrani ime LED traku *StripName*, objekte ki sestavljajo trak v formatu JSON *AnimationObjectsJSON*, število diod na traku *NumberOfDiodes*, referenco na projekt *ProjectId* in enolično oznako *StripId*.



Slika 2.5: Diagram podatkovnega modela

## 2.4 Implementacija uporabniškega dela

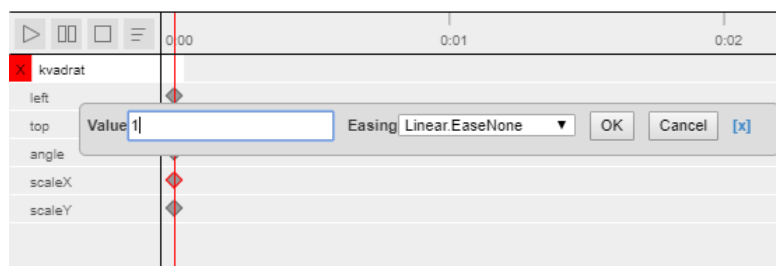
Implementacija uporabniškega dela se je začela z osnovno postavitvijo strani projekta. Naredili smo dva zavihka s prostorom za platno in orodno vrstico. Večinski del uporabniškega dela je bilo grajenje funkcionalnosti v Javascriptu. Uporabili smo knjižnico Fabric.js zaradi že implementiranega osnovnega izbiranja in nadzorovanja objektov po platnu, ki smo ga uporabili pri manipulaciji diod. V prvem zavihku smo implementirali funkcijo za ustvarjanje novega LED traku. LED trak smo predstavili s krogi, ki predstavljajo posamezne diode, in so med sabo povezani s črtami. Na istem zavihku smo uvedli še shranjevanje z asinhronim klicem na strežnik, ki pošlje vse trenutne oblike na platnu v formatu JSON. Dodali smo še funkcionalnost, ki trenutno izbrani trak postavi v poljubno število vrstic. Na Sliki 2.6 je prikazan končni izgled prvega zavihka projekta.



Slika 2.6: Zaslonski posnetek zavihka za oblikovanje LED traku

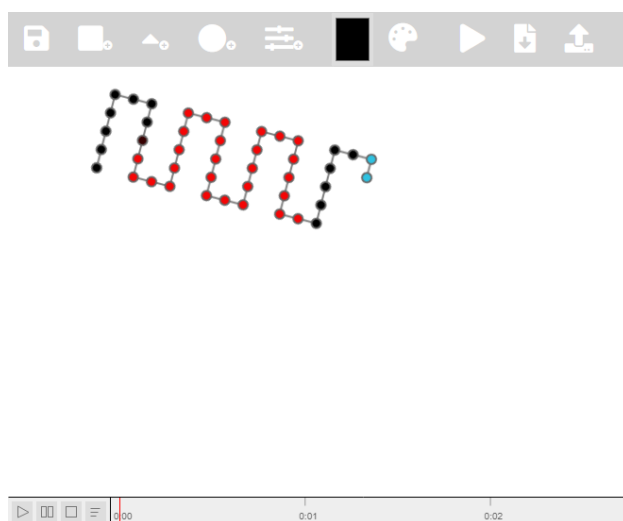
V drugem zavihku smo implementirali animacijo oblik s katerimi smo določali barvo izbranega traku. Uporabili smo knjižnico Timeline.js, ki implementira uporabniški vmesnik za ustvarjanje animacij s ključnimi okvirji (ang. keyframe animation). Princip take animacije je, da se objektu, ki ga hočemo animirati, določi nek začetni in končni ključni okvir, ki vsebuje njegovo pozicijo na nekem začetnem in končnem času. Vmesni okvirji (ang. inbetweens) se izračunajo na podlagi začetnih dveh pozicij. Okvirji skupaj dajejo vtis gladkega premikanja. Oblike dodajamo na platno s pritiskom na

gumb v orodni vrstici, pojavi se nam pojavno okno, v katerem poimenujemo dodano obliko. Na platnu se nam izriše oblika in doda v časovnico, kjer se ji s parametri ključnih okvirjev določi privzeto postavitev. Nastavi se pet parametrov in sicer X in Y koordinate oblike na platnu, kot (pod katerim je postavljena oblika), skala širine in skala dolžine oblike (Slika 2.7). S klikom na posamezen parameter lahko spremenimo njegovo vrednost in tip translacije, lahko pa ga odstranimo s pritiskom na križec. Nove parametre lahko dodamo z dvojnimi klikom na poljubni čas in lastnost posamezne oblike v časovnici. Drugi način je, da postavimo obliko v poljubno pozicijo in ob pritisku na gumb “Set properties” (peti zaporedni gumb v orodni vrstici na Sliki 2.8) se za trenutno določen čas nastavijo ustrezni parametri. Animacijo predmetov predvajamo in ustavimo s pomočjo kontrolnih gumbov v zgornjem levem kotu časovnice.



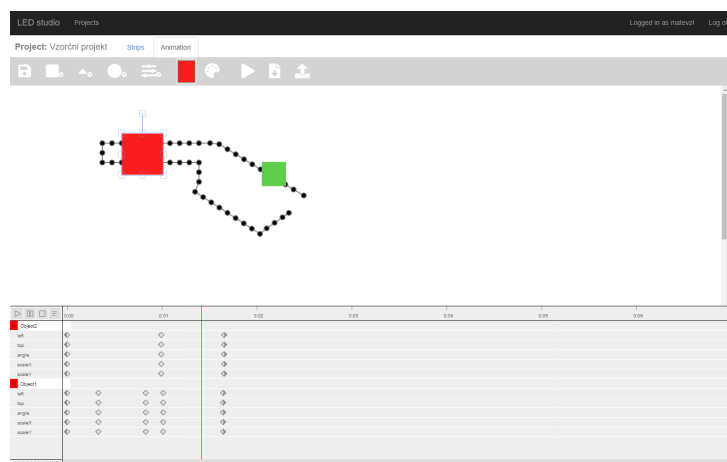
Slika 2.7: Časovnica z enim objektom in eno začetno pozicijo

Na platnu je hkrati prikazan tudi trenutno izbran LED trak, nad katerim se bo izvedla animacija. Animacija traku je določena z barvami in pozicijami oblik. Posamezna dioda je v nekem trenutku ugasnjena, če je ne prekriva nobena oblika. Če dioda v nekem trenutku prekriva oblika, se ji določi barva oblike. Za bolj nazoren pregled animacije, takšne kot se bo izvedla na fizičnih diodah, je na voljo funkcionalnost predogleda animacije. Ob kliku na gumb “Preview” (sedmi zaporedni gumb v orodni vrstici na Sliki 2.8) se začasno skrijejo oblike na platnu in nad trakom se izvede animacija tako, da se po-barvajo ustrezne diode (Slika 2.8).



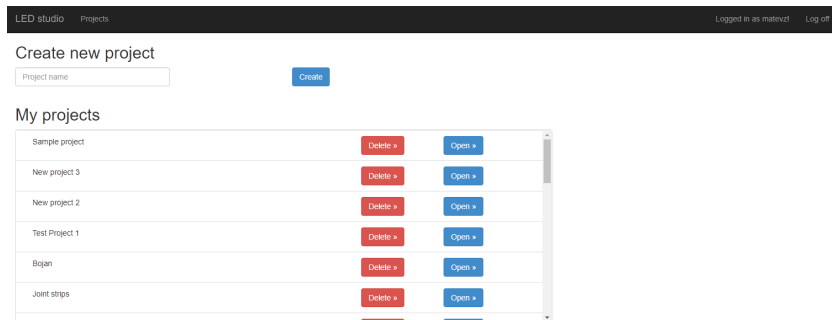
Slika 2.8: Izgled LED traku med izvajanjem predogleda

Barvo oblike določimo tako, da izberemo obliko na platnu in kliknemo na gumb poleg barvnega polja v orodni vrstici. Obliki se nastavi barva, ki je trenutno izbrana v barvnem polju. Če hočemo barvno polje spremeniti, nanj kliknemo in odpre se nam dialog za izbiro barv. Končen izgled zavihka za ustvarjanje aplikacij je prikazan na Sliki 2.9.



Slika 2.9: Zaslonski posnetek zavihka, za ustvarjanje animacij

Na Sliki 2.10 je prikazan še končni izgled strani, kjer lahko uporabnik ustvari nov projekt, ali pa odpira in odstranjuje že ustvarjene projekte, ki so zbrani v seznamu.



Slika 2.10: Zaslonski posnetek strani, kjer uporabnik upravlja z projekti.





## Poglavje 3

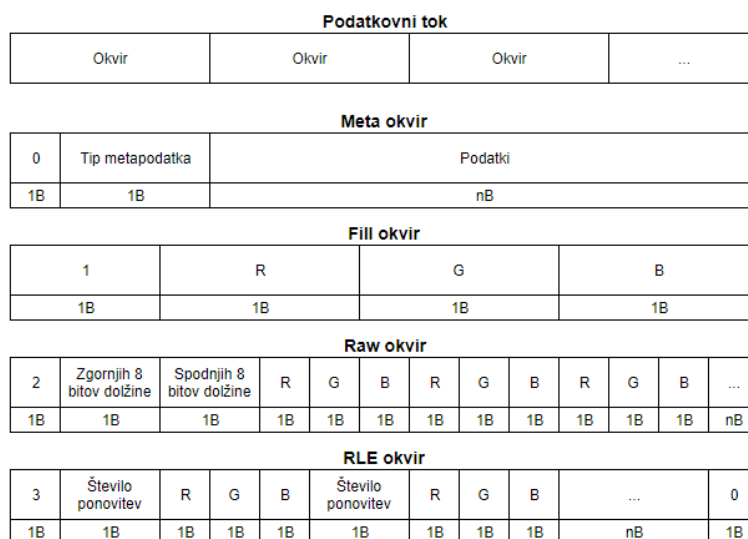
# Krmiljenje LED traku

Mikrokrmilniki imajo na voljo zelo malo pomnilnika, zato smo ustvarili format, ki nestisnjne podatke animacije zapiše v bolj učinkovito obliko. Zasnovan je tako, da ga je v prihodnje mogoče dopolniti z novimi metodami za stiskanje podatkov, implementirana je zgolj začetna različica. Na strani mikrokrmilnika smo potem implementirali dekodiranje zastavljenega formata in zapis animacije na fizični LED trak.

### 3.1 Sestava formata animacije

Format smo razdelili v skupine podatkov, ki jih bomo poimenovali okvirji (Slika 3.1 - Podatkovni tok). Zastavili smo štiri tipe okvirjev. Prvi tip okvirja služi za pošiljanje metapodatkov. Drugi tip okvirja je najbolj enostaven, vsebuje podatke ene barve, s katero zapolni vse diode na traku. Tretji tip okvirja vsebuje zaporedje vseh barv za vsako posamezno diodo. Zaradi omejenega prostora, ki ga ima krmilnik, smo implementirali še en okvir, ki vsebuje podatke barv za vsako diodo, nad katerimi je bil izvedeno kodiranje s pomočjo zaporedij (ang. Run-length encoding - RLE). Kodiranje smo izvedli nad vrednostmi RGB, če se je ista vrednost barve pojavila n-krat, smo to namesto celotnega zaporedja zapisali število ponovitev in nato vrednosti RGB. Vsak okvir se začne z enim bajtom, ki označuje, za kateri tip gre:

- **Meta** okvir (Slika 3.1 - Meta okvir) označujemo z bajtom 0. Sledi še en bajt, ki označuje tip metapodatka. Zaenkrat smo definirali le en tip metapodatka z oznako 0, ki vsebuje informacijo o številu okvirjev na sekundo v dveh bajtih. Okvir še ni v implementiran in je namenjen uporabi v prihodnosti.
- **Fill** okvir (Slika 3.1 - Fill okvir) označujemo z bajtom 1. Ta tip okvirja vsebuje le še tri dodatne bajte, ki označujejo barvo z vrednostmi RGB. Ta barva zapolni celotni trak.
- **RAW** okvir (Slika 3.1 - RAW okvir) označujemo z bajtom 2. Sledita mu dva bajta, ki predstavljata dolžino traku v številu diod. Potem se zapiše le še zaporedje vrednosti RGB.
- **RLE** okvir (Slika 3.1 - RLE okvir) označujemo z bajtom 3. Sledi zaporedje štirih bajtov. Prvi bajt označuje število ponovitev posamezne barve, naslednji trije pa se uporabljajo za vrednosti RGB te barve. Konec okvirja se označi z zapisom bajta z vrednostjo 0.

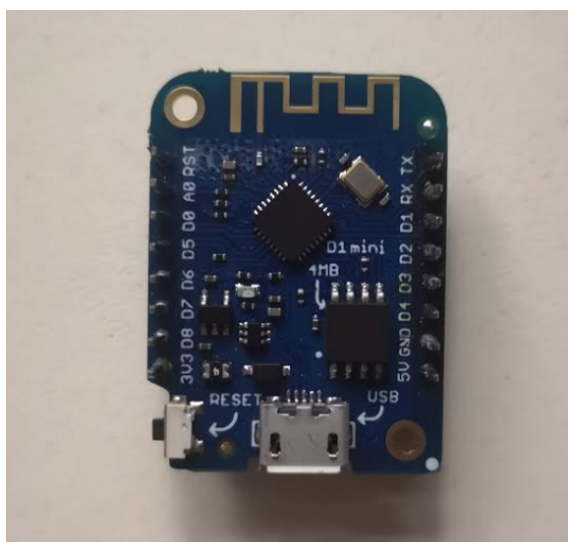


Slika 3.1: Shema strukture formata za shranjevanje animacij

Pri sestavi formata iz okvirjev, se za uporabo določenega tipa okvirja odločimo glede na prostor, ki ga zavzame. Uporabi se okvir, ki zavzame najmanjše število bajtov. Format smo zasnovali tako, da ga je v prihodnje mogoče dopolniti z novimi metodami za stiskanje podatkov, to je zgolj začetna različica.

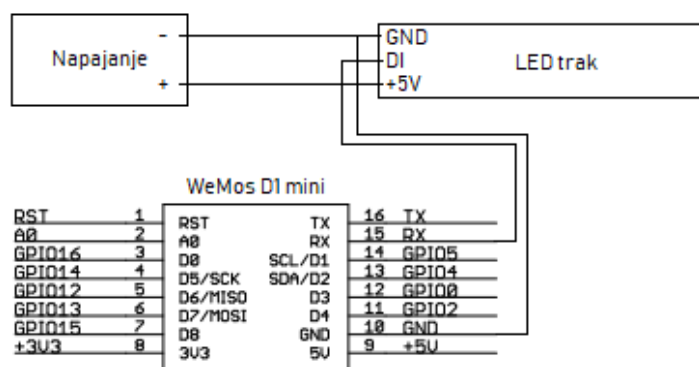
## 3.2 Krmiljenje

Za krmiljenje LED traku smo uporabljali ploščico WeMos D1 mini V3.0.0 [13] podjetja WeMos (Slika 3.2), ki je zasnovan na modulu ESP-12S podjetja Ai Thinker, centralna komponenta tega modula pa je ESP8266 čip podjetja Espressif. Čip ESP8266 ima integriran celoten modem za Wi-Fi, ki je zmožen komunikacije preko 802.11 b/g/n standardov Wi-Fi [3], sam ESP-12S modul pa vsebuje anteno Wi-Fi. Ta omogoča povezavo na Wi-Fi omrežje in ustvarjanje enostavnih povezav TCP/IP. Podatkovno krmiljenje samega traku poteka preko protokola I2S, ki je namenjen za povezavo digitalnih avdio naprav.



Slika 3.2: WeMos D1 mini V3.0.0

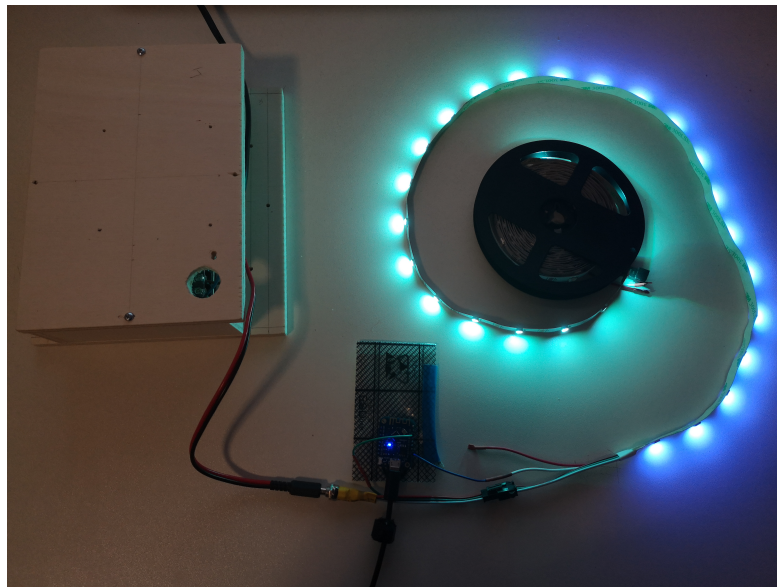
Uporabljali smo LED trak SK6812RGBW, ki uporablja module SMD LED tipa 5050, in je eden izmed bolj razširjenih modulov. Številke tipa modula nam povedo dimenzije diode, v tem primeru gre za diodo z velikostjo 5,0 mm x 5,0 mm. Trak potrebuje napajanje napetosti od 3,5 V do 5,5 V. Posamezni modul vsebuje tri majhne diode, in sicer rdečo, zeleno in modro. Na Sliki 3.4 je prikazan osvetljen trak, skupaj z mikrokrmilnikom in napajalnikom. Za trak smo priskrbeli zunanje napajanje, mikrokrmilnik pa se napaja preko vhoda micro USB. LED trak prejme poleg napajanja še podatke na podatkovni vhod, ki smo ga povezali z razvojno ploščico. Shema celotnega vezja je prikazana na Sliki 3.3.



Slika 3.3: Priključitev LED traku

Razvijali smo v programskem okolju Arduino IDE, ki je namenjen programiranju razvojnih ploščic. Za delo z modulom ESP8266 smo uporabili tudi dodatno knjižnico [4]. Mikrokrmilnik se poveže na vnaprej določeno omrežje Wi-Fi in čaka, da dobi podatke preko povezave TCP/IP. Podatke si zapišemo v krožni pomnilnik (ang. circular buffer) in jih z ustrezno hitrostjo dekodiramo. Ker mikrokrmilnik nima na voljo veliko prostora, izkoriščamo mehanizem upravljanja hitrosti prenosa podatkov (ang. flow control) protokola TCP. Ta poskrbi, da prejemnik prejme le toliko podatkov, kot jih lahko obdelata. V dekodirju razčlenimo podatke glede na tipe okvirjev, ki smo jih

predhodno definirali. Uporaba okvirja z metapodatki na strani krmilnika trenutno še ni implementirana. Dekodirajo se okvirji tipa Fill, RAW in RLE. Dekodirane podatke pošljemo na vodilo I2S z določeno vzorčno frekvenco. Ko se komunikacija z odjemalcem konča, je krmilnik pripravljen na sprejem novih podatkov.



Slika 3.4: LED trak priklopljen na mikrokrmilnik



## Poglavje 4

# Študija uporabniške izkušnje

Da bi lahko bolje ovrednotili našo aplikacijo, smo za njo izvedli študijo uporabniške izkušnje. Za kandidate smo izbrali nekaj oseb, ki so v računalniških vodah, in nekaj oseb, ki s programiranjem nimajo posebnih izkušenj. Študijo smo izvedli v štirih delih. V prvem delu smo kandidatu predstavili aplikacijo. Povedali smo mu, čemu je namenjena, kakšna je bila naša motivacija za njen razvoj, in omenili nekaj primerov rabe aplikacije. Nato smo kandidatu pokazali, kakšne funkcionalnosti aplikacija podpira, in kako jih uporabljati. V drugem delu smo kandidatu zadali tri naloge s ciljem, ki ga je moral doseči z uporabo aplikacije. Ko je oseba končala z nalogami, je v okviru tretjega dela študije izpolnila kratek vprašalnik, v katerem je ovrednotila trditev po Likertovi lestvici [1] od ena (se popolnoma ne strinjam) do pet (se povsem strinjam). Na koncu je sledil še intervju, kjer smo vsakemu uporabniku zastavili štiri vnaprej določena vprašanja in glede na podan odgovor postavili še kakšna dodatna podvprašanja, da so lahko čim bolj izjasnili svoja mnenja.

### 4.1 Naloge uporabniku

Uporabnikom smo zastavili tri naloge:

#### 1. Naloga 1

- V aplikaciji se registrirajte kot nov uporabnik.

- Vpišite se v aplikacijo z novo ustvarjenim računom.
- Ustvarite nov projekt, in ga poimenujte kot “Naloga 1”.
- V projektu ustvarite nov LED trak s tridesetimi diodami in poljubnim imenom.
- Animirajte trak tako, da bo skupek diod rdeče barve v eni sekundi prepotoval z leve strani proti desni, in skupek diod zelene barve, prepotoval z desne strani proti levi.
- Animacijo prenesite na fizični LED trak.

## 2. Naloga 2

- Ustvarite nov projekt z imenom “Naloga 2”.
- Ustvarite nov trak s tridesetimi diodami, in ga postavite v tri vrstice.
- Animirajte trak tako, da bodo v poljubnem času skozi vsako vrstico diod prepotovale barve, ki so različne za vsako vrstico.
- Animacijo prenesite na fizični LED trak.

## 3. Naloga 3

- Ustvarite nov projekt z imenom “Naloga 3”.
- Ustvarite nov trak s tridesetimi diodami.
- Postavite trak v obliko črke M.
- Ustvarite tako animacijo, da bo polovica črke M svetila z modro barvo in druga polovica z zeleno barvo.
- Animacijo prenesite na fizični LED trak.

## 4.2 Anketa

Anketa je sestavljena iz sedmih trdilnih vprašanj. Uporabniki so trditev ovrednotili z ocenami od 1 (se popolnoma ne strinjam) do 5 (se povsem strinjam).



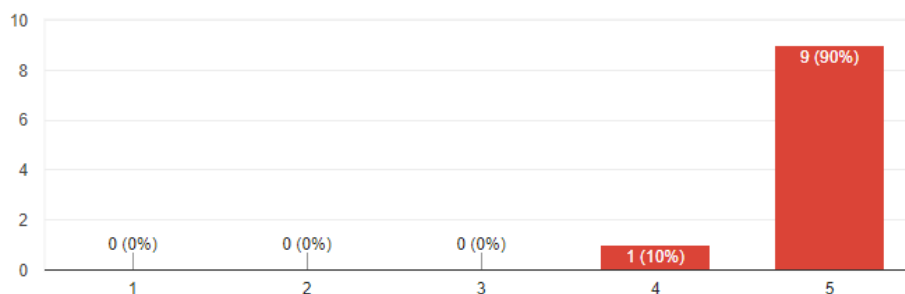
1. Aplikacija je zanimiva za ljudi, ki se ljubiteljsko ukvarjajo Z LED trakovi.
2. Pri navigiranju skozi aplikacijo nisem imel težav.
3. Ustvarjanje LED trakov je enostavno in intuitivno.
4. Pri ustvarjanju trakov nisem imel težav.
5. Ustvarjanje animacij je intuitivno.
6. Pri ustvarjanju animacij nisem imel težav.
7. Aplikacija je delovala zadovoljivo, brez performančnih težav.

### 4.3 Rezultati ankete

Anketirali smo deset oseb, od tega do bile tri ženske in sedem moških.

1. **Aplikacija je zanimiva za ljudi, ki se ljubiteljsko ukvarjajo z LED trakovi.**

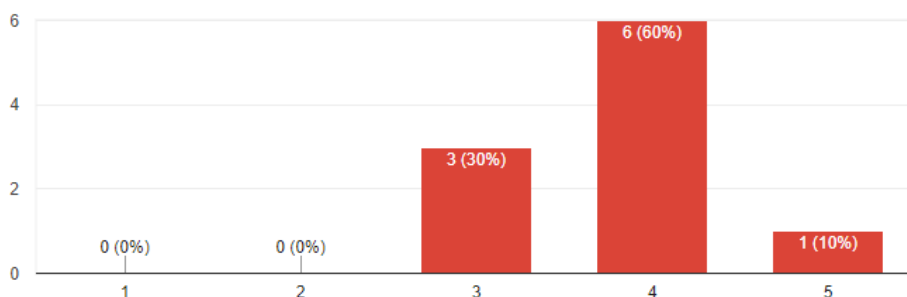
Iz rezultatov je razvidno (Slika 4.1), da se vsi anketiranci strinjajo s trditvijo. Podoben rezultat smo tudi pričakovali z ozirom, druge možnosti upravljanja trakov, ki so uporabnikom trenutno na voljo.



Slika 4.1: Rezultati prve trditve ankete. Število odgovorov na osi y in vrednost odgovora na osi x.

## 2. Pri navigiranju skozi aplikacijo nisem imel težav.

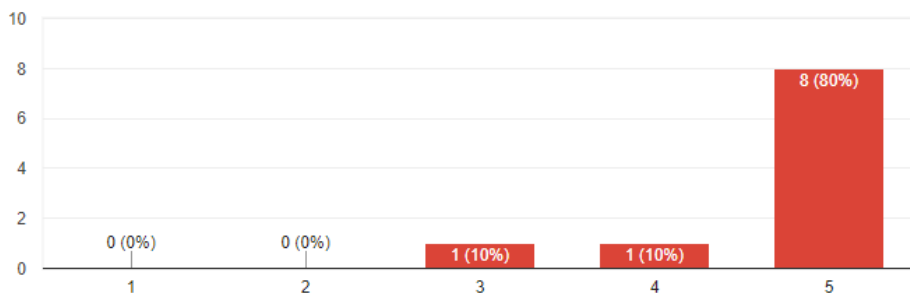
Rezultati (Slika 4.2) prikazujejo, da čeprav so uporabniki v povprečju čez aplikacijo navigirali dovolj dobro, še vedno obstajajo možnosti izboljšanja, glede na to, da se je s trditvijo popolnoma strinjala le ena oseba.



Slika 4.2: Rezultati druge trditve ankete. Število odgovorov na osi y in vrednost odgovora na osi x.

## 3. Ustvarjanje LED trakov je enostavno in intuitivno.

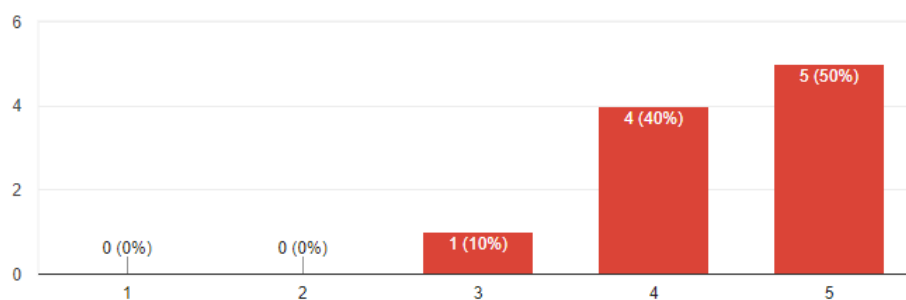
Z rezultati te trditve (Slika 4.3) smo zadovoljni. Podobne smo tudi pričakovali, saj proces ustvarjanja traku zahteva dokaj malo klikov, zato je tudi preprosto za uporabo.



Slika 4.3: Rezultati tretje trditve ankete. Število odgovorov na osi y in vrednost odgovora na osi x.

#### 4. Pri ustvarjanju trakov nisem imel težav.

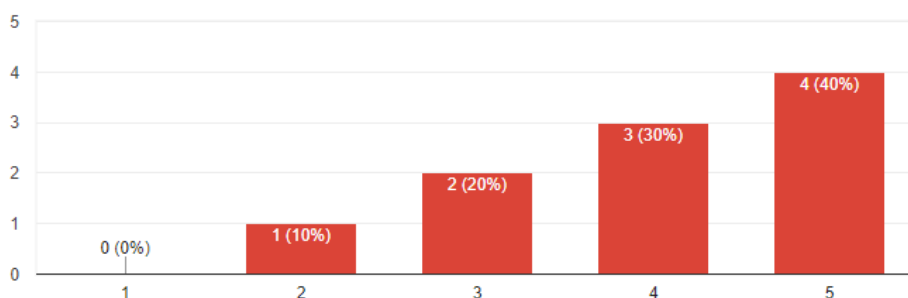
Pričakovali smo, bodo rezultati te trditve (Slika 4.4) podobni rezultatom prejšnje trditve. Dejanski rezultati so slabši od pričakovanih. Med opazovanjem kandidatov pri izpolnjevanju nalog smo opazili, da so imeli uporabniki manjše težave z izbiranjem posameznih diod, ker so dokaj majhne.



Slika 4.4: Rezultati četrte trditve ankete. Število odgovorov na osi y in vrednost odgovora na osi x.

#### 5. Ustvarjanje animacij je intuitivno.

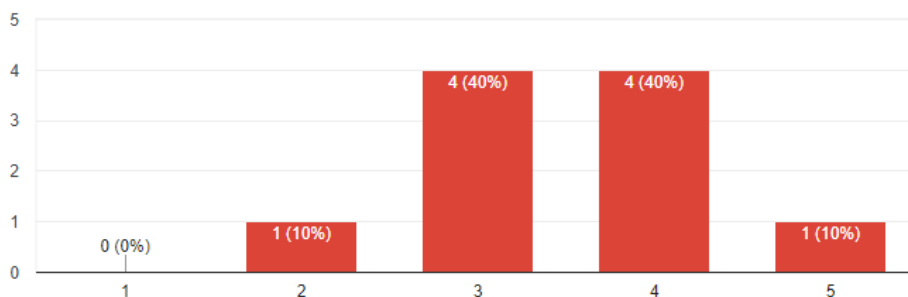
Pričakovali smo, da se bo pri tej funkcionalnosti aplikacije pojavila razlika med uporabniki, ki so že kdaj ustvarjali animacije s ključnimi okvirji in tistimi, ki to počnejo prvič. Rezultati (Slika 4.5) potrjujejo, da ta funkcionalnost ni intuitivna za vsakega uporabnika, saj so rezultati dokaj razpršeni, vseeno pa povprečje kaže, da je funkcionalnost bolj intuitivna kot neintuitivna.



Slika 4.5: Rezultati pete trditve ankete. Število odgovorov na osi y in vrednost odgovora na osi x.

#### 6. Pri ustvarjanju animacij nisem imel težav.

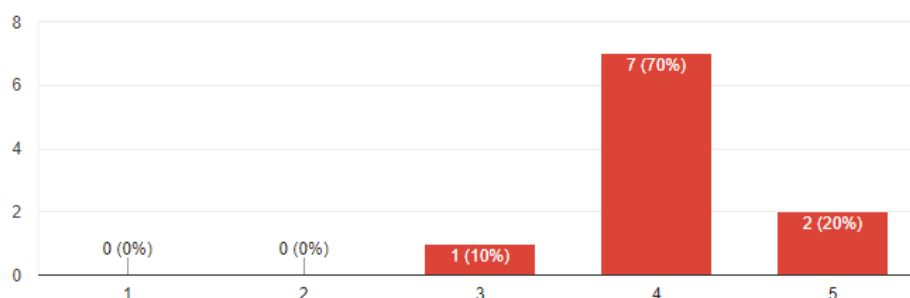
Rezultati te trditve (Slika 4.6) so nekoliko presenetljivi, saj smo pričakovali rezultate, ki bi bili podobni rezultatom prejšnje trditve. Razvidno je torej, da so imeli uporabniki pri tej funkcionalnosti največ težav in obstaja še veliko prostora za izboljšanje.



Slika 4.6: Rezultati šeste trditve ankete. Število odgovorov na osi y in vrednost odgovora na osi x.

#### 7. Aplikacija je delovala zadovoljivo, brez performančnih težav.

Rezultati (Slika 4.7) kažejo, da je aplikacija v večini delovala tako, kot je bilo namenjeno in uporabnikom ni povzročala večjih performančnih težav.



Slika 4.7: Rezultati sedme trditve ankete. Število odgovorov na osi y in vrednost odgovora na osi x.

## 4.4 Intervju

1. Kakšne so vaše kritike aplikacije, kaj vam ni bilo všeč in se vam zdi da bi bilo lahko izvedeno boljše?
  - Intuitivnost pri animaciji je največja težava. Nastavitve položajev bi urejal preko dvojnega klika na izbrano obliko, urejanje lastnosti v časovnici bi odstranil. Popravil bi tudi orodno vrstico, zdi se mi, da je preveč gumbov, postavil bi jih v skupine, gumbе za dodajanje različnih oblik pa združil v enega. Pri izbiranju barve mislim, da je potrebnih preveč klikov.
  - Ko postaviš obliko v prostor, bi dodal možnost, da izbereš več parametrov naenkrat in jih izbrišeš, ne pa vsakega posebej. Sedaj se pokaže kratko ime funkcionalnosti, ko na gumb pokažeš z miško, lahko bi imel za prvo uporabo poleg imena še pojavna okna, ki bolj podrobno razložijo posamezno funkcionalnost.
  - Zmotilo me je brisanje parametrov, rada bi vse označila in izbrisala naenkrat. Manjka funkcija "undo". Ko sem v časovnici delala z objekti, se je zgodilo, da okna nisem mogla povsem pomakniti do dna, in sem ga morala razširiti navzgor. Spreminjanje barv mi je vzelo preveč klikov, dodala bi več barv že v orodno vrstico.

- Nerodno je bilo avtomatsko postavljanje oblike v začetno pozicijo, raje bi, da je oblika na začetku brez pozicij. Nerodno se mi je zdelo tudi eksplicitno shranjevanje oblike v prvem zavihku, da se je posodobila v drugem zavihku. Pričakoval bi, da bi se to zgodilo avtomatsko ob zamenjavi zavihka.”
  - Nerodno mi je bilo brisanje posameznih parametrov.
  - Moja ena in edina kritika je brisanje pozicij na časovnici, rada bi jih brisala več naenkrat, posamezno brisanje se mi zdi zelo zamudno.
  - Prvič, dodal bi izbiranje več parametrov naenkrat, da jih lahko zbršem z enim klikom. Tudi bolj nazorno bi označil vrh črte, ki označuje trenutni čas na časovnici, da je bolj očitno, da jo moraš potegniti iz vrha. Dodal bi še neke začetniške korake, ki ti ob prvi uporabi razložijo, kako aplikacijo uporabljati.
  - Pozicije na časovnici, bi se lahko prepisale z novimi, da jih ne bi rabil brisati. Mogoče bi tudi malo izboljšal dizajn aplikacije.
  - Brisanje pozicij na začetku bi bilo lahko boljše narejeno. Določanje barve bi poenostavil, rad bi kliknil na obliko, in ji takoj določil barvo. Združil bi predogled animacije na traku in predvajanje animacije objektov.
  - Ko se doda nova oblika, bi odstranil to, da ti jo postavi v neko privzeto pozicijo.
2. Katere so po vašem mnenju dobre strani te aplikacije, kaj se vam je zdelo dobro izvedeno?
- Zdi se mi dobro, da lahko isto animacijo porabim za več različnih trakov.
  - Postavljanje trakov mi je všeč, tudi animacija se mi zdi logična, postavljanje začetnih in končnih pozicij dela kot pričakovano.

- Na začetku se mi zdi zelo dobro, da se posamezne diode lahko poljubno postavljajo po prostoru, da se ne mučiš s predstavo animacije na ravnem traku.
- Sigurno mi je bilo všeč, da sem imel proste roke pri definiranju oblike traku. Všeč mi je bil tudi sam princip animacije ... da postaviš začetno in končno pozicijo.
- Oblikovanje traka mi je bilo zelo smiselno, pa tudi animacija.
- Čisto na splošno se mi zdi, da je zelo enostavno zastavljena, ni preveč nekih ikon, nastavitvev, človek se lahko znajde brez nekih dodatnih navodil.
- Pri ustvarjanju trakov nimam nobenih pripomb.
- Zelo enostavno, ni veliko komand, takoj je razvidno, kako se kaj naredi.
- Definiranje oblik trakov mi je bilo všeč. Tudi koncept animiranja barv z oblikami je hiter, na začetku sem si predstavljal, da bom moral klikati vsako diodo posebej. Preprosta je za uporabo.
- Aplikacija se mi zdi enostavna za uporabo, postavljanje oblik v začetne in končne pozicije se mi zdi dober način animiranja.

3. Imate kakšne ideje, kako bi lahko aplikacijo v prihodnosti nadgradili?

- Mislim, da bi bila aplikacija veliko bolj uporabna, če ne bi bila spletna, ampak mobilna. Uporabljam že mobilno aplikacijo za upravljanje pametnih žarnic, če bi bila ta spletna, je ne bi uporabljal.
- Naslednja stopnja je postavljanje trakov v 3D prostor, na primer ovijanje traku okoli objekta in predogled v 3D prostoru.
- Kopiranje objektov in njihovih pozicij s spremenjeno barvo. Dodala bi avtomatsko postavljanje traku v več privzetih oblik, na primer zvezdica, smrekica, krogec.

- Rad bi združeval trakove, dodal bi več trakov naenkrat na polje. Aplikacijo bi lahko tudi dopolnil v neke vrste market, da si ljudje med seboj lahko delijo oblike in animacije. Mogoče bi lahko naložil neko slikico, kot konturo, in bi aplikacija to spremenila v obliko LED traku.
  - Nimam idej.
  - Dodala bi predolge oblik, ki se jih lahko uporabi za trakove. Dodala bi tudi možnost dodajanja glasbe v časovnico, da bi lahko ustvarjala animacije, ki so v skladu z glasbo.
  - Dobro bi bilo, če bi pri animaciji lahko izbral posamezne diode, in jim določil barvo direktno. Mogoče dodajanje več trakov naenkrat. Ko se oblike prekrijejo, bi se barve lahko seštele.
  - Imel bi že neke vnaprej pripravljene animacije, ki bi jih lahko uporabil na traku.
  - Več trakov naenkrat. Rad bi videl tudi, kakšne vnaprej definirane oblike. Mogoče bi imel možnost prostega risanja, kot v slikarju, in nato bi se trak postavil v narisano obliko.
  - Dodal bi še možnost, da lahko ena oblika čez čas spreminja barve, da ne rabiš dodajati vedno nove oblike, ko hočeš drugo barvo.
4. Če bi želeli ustvariti animacijo na takem traku, bi raje uporabili to aplikacijo, ali bi se lotili programiranja posameznih diod na nižjem nivoju?
- Aplikacijo, v programiranje se ne bi spuščal.
  - Programiranje po mojih izkušnjah vzame preveč časa, definitivno bi uporabil aplikacijo.
  - Aplikacijo.
  - Ker vem, kako je programirati nizkonivojsko z mikrokrmilniki, bi definitivno raje uporabil aplikacijo, zdi se mi nedvomno najhitrejša opcija.



- Uporabila bi aplikacijo.
- Poslužila bi se aplikacije, definitivno.
- Sigurno se ne bi spravil programirati, izbral bi aplikacijo.
- Z veseljem bi uporabil vašo aplikacijo.
- To [aplikacija] je sigurno boljše od programiranja.
- Sigurno je hitreje uporabiti aplikacijo.

Iz intervjujev je razvidno, da imamo kar nekaj povratnih informacij, ki se prekrivajo. Ena najpogostejših kritik je bila brisanje privzetih parametrov ključnih okvirjev. Uporabnikom ni bilo všeč, da so parametre morali brisati vsakega posebej. To se tudi sklada z rezultati ankete, ki kažejo, da so imeli uporabniki z animacijo nekaj težav. Rešitev tega problema je jasna in tehnično nezahtevna. Implementiral bi izbiro več parametrov naenkrat in omogočili brisanje s pritiskom na gumb. Poleg tega bi lahko tudi odstranili določitev privzete pozicije oblike, in s tem zmanjšali število parametrov na časovnici. Funkcija, s katero so bili uporabniki najbolj zadovoljni, je definiranje oblike traku. Všeč jim je bilo prosto postavljanje posameznih diod v zelen položaj. Prejeli smo zelo veliko zanimivih idej za nadgradnjo aplikacije, največkrat zaželeno pa je bilo delo z več trakovi naenkrat. Uporabniki so bili z aplikacijo na splošno zadovoljni, na njeno uporabnost tudi kaže dejstvo, da je vsak kandidat odgovoril, da bi se izognil delu z mikrokrmilnikom in uporabil aplikacijo.



# Poglavje 5

## Sklep in nadaljne delo

V sklopu diplomske naloge smo želeli razviti spletno aplikacijo, ki bi uporabnikom omogočala ustvarjanje animacij po meri za pametne LED trakove. Nastala je aplikacija, ki so jo uporabniki sprejeli kot boljšo alternativo nizkonivojskemu programiranju trakov.

### 5.1 Sklep

Iz rezultatov ankete in opravljenih intervjujev je moč razbrati, da so imeli uporabniki z aplikacijo povečini dobro izkušnjo. Kritike, ki smo jih prejeli, v splošnem zahtevajo popravke, ki niso tehnično težki, in se večinoma lahko brez težav implementirajo. Uporabnikom je bila predvsem všeč prostost pri oblikovanju traku in princip animacije z začetnimi in končnimi pozicijami oblik.

### 5.2 Nadaljnje delo

Za aplikacijo smo prejeli veliko različnih idej za razširitev in nadgradnjo. Ena izmed jasnih razširitev je delo z več trakovi naenkrat. Gre tudi za eno izmed bolj zanimivih razširitev, saj omogoča sestavo zaslona iz več pametnih LED trakov. Zanimiva je tudi umestitev LED traku v tridimenzionalni prostor.

To bi omogočilo bolj nazorno in jasno predstavitev traku v fizičnem prostoru. Še ena želja po nadgradnji, ki je bila večkrat izražena s strani uporabnikov, je avtomatična postavitev traku v več vnaprej definiranih oblik, ki so pogosto v uporabi. Predlagana je bila tudi vključitev glasbe, to bi bilo poleg ročne sinhronizacije z glasbo mogoče implementirati tudi samodejno, z uporabo hitre Fourierjeve transformacije zvočnih signalov [7]. Zelo uporabno bi bilo tudi prostoročno risanje oblike traku in zaznavanje oblike traku iz naložene slike. Treba bi bilo tudi izboljšati direktni prenos animacije iz aplikacije. Za namene uporabniške študije je bilo implementirano pošiljanje podatkov na vnaprej določen naslov IP s strani strežnika, ki se je izvajal lokalno. Iz aplikacije, ki je objavljena na spletu, pa se prenese datoteko z animacijo, ki se jo potem lahko lokalno pošlje preko namenskih orodij (npr. netcat). V prihodnje bi to lahko rešili z uporabo namenskega vtičnika za brskalnik, ki bi poskrbel za lokalno pošiljanje podatkov na mikrokrmilnik. Odkrivanje mikrokrmilnikov na lokalni mreži, bi lahko rešili s protokolom multicast DNS [2].

Zagotovo pa je prvi korak sprememba že implementiranih funkcionalnosti glede na kritike uporabnikov in šele nato nadgrajevanje z novimi.

# Literatura

- [1] Allen, I Elaine and Seaman, Christopher A. Likert scales and data analyses. *Quality progress*, 40(7):64–65, 2007.
- [2] S. Cheshire and M. Krochmal. Multicast DNS, February 2013. RFC 6762, DOI 10.17487/RFC6762.
- [3] ESP8266. Dosegljivo: <https://www.ai-thinker.com/product/esp8266/>. [Dostopano: 25. 08. 2018].
- [4] ESP8266 WiFi knjižnica. Dosegljivo: <https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/server-class.html>. [Dostopano: 25. 08. 2018].
- [5] Fabric.js. Dosegljivo: <http://fabricjs.com/>. [Dostopano: 25. 08. 2018].
- [6] G. Curcio and L. Piccardi and F. Ferlazzo and A. M. Giannini and C. Burattini and F. Bisegna. LED lighting effect on sleep, sleepiness, mood and vigor. In *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, pages 1–5, June 2016.
- [7] Kulovic, Jože. Sistem za prikaz predvajane glasbe s pomočjo led diod, 2013.
- [8] LED Strip Studio. Dosegljivo: <https://ledstripstudio.com/faq/>. [Dostopano: 25. 08. 2018].

- 
- [9] Meschtscherjakov, Alexander and Döttlinger, Christine and Rödel, Christina and Tscheligi, Manfred. ChaseLight: Ambient LED Stripes to Control Driving Speed. In *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '15, pages 212–219, New York, NY, USA, 2015. ACM.
- [10] S. Song and S. Yamada. Investigating effects of light animations on perceptions of a computer: Preliminary results. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 17–22, Aug 2017.
- [11] Schiller, B. *The Automated Lighting Programmer's Handbook*, pages 137–138. Taylor & Francis, 2003.
- [12] Timeline.js. Dosegljivo: <http://marcinignac.com/blog/timeline-js/>. [Dostopano: 25. 08. 2018].
- [13] WeMos D1 mini V3.0.0. Dosegljivo: [https://wiki.wemos.cc/products:d1:d1\\_mini](https://wiki.wemos.cc/products:d1:d1_mini). [Dostopano: 25. 08. 2018].