



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VÝSKUM A VÝVOJ UI/UX PRE KOMENTOVANIE ŠPORTOVÝCH POZÍCIÍ

RESEARCH AND DEVELOPMENT OF UI/UX FOR COMMENTING SPORTS POSES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN BARBORÍK

VEDOUcí PRÁCE

SUPERVISOR

prof. Ing. ADAM HEROUT, Ph.D.

BRNO 2024

Zadání bakalářské práce



155252

Ústav: Ústav počítačové grafiky a multimédií (UPGM)
Student: **Barborík Martin**
Program: Informační technologie
Název: **Výzkum a vývoj UI/UX pro komentování sportovních pozic**
Kategorie: Uživatelská rozhraní
Akademický rok: 2023/24

Zadání:

1. Seznamte se s problematikou vybraného sportovního tréninku a jeho podpory pomocí digitálních aplikací.
2. Seznamte se s problematikou výzkumu a vývoje UX/UI.
3. Proveďte výzkum preferencí a zájmů potenciálních uživatelů. Formulujte požadavky na UX/UI pro komentování sportovních pozic.
4. Prototypujte dílčí prvky UI, testujte je s vhodnými respondenty a iterativně je vylepšujte.
5. Integrujte dílčí prvky UI do komplexnějšího UI pro komentování sportovních pozic.
6. Testujte vyvíjené UI a iterativně je vylepšujte.
7. Vytvořte demonstrační UI umožňující předvést a vyhodnotit dosažené výsledky. Formulujte závěry a doporučení.
8. Zhodnoťte dosažené výsledky a navrhňte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

Literatura:

- Elisa Păduraru, Fundamentals of Creating a Great UI/UX, Creative Tim, 2022
- Tidwell et al.: Designing Interfaces: Patterns for Effective Interaction Design, O'Reilly, 2020
- Steve Krug: Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, ISBN: 978-0321965516
- Steve Krug: Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability, ISBN: 978-0321657299

Při obhajobě semestrální části projektu je požadováno:
položky 1, 2, 3, značné rozpracování položky 4

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Herout Adam, prof. Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, prof. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 9.5.2024
Datum schválení: 9.11.2023

Abstrakt

Cielom tejto bakalárskej práce je výskum a iteratívny vývoj užívateľského rozhrania pre vizuálne komentovanie jogových pozícií. Tento proces hlboko závisel od výskumu s potenciálnymi užívateľmi. Boli zozbierané sady okomentovaných fotografií na papieroch. Z ktorých sa zaujímavé a opakované prvky aplikovali do vyvíjaného rozhrania. Priebežne sa opakovane vykonávali kvalitatívne testy, z pozorovaní som našiel nedostatky, aplikoval zmeny a testoval znovu. Výsledkom práce je demonštračné užívateľské rozhranie realizované ako progresívna webová aplikácia.

Abstract

The goal of this bachelor's thesis is the research and iterative development of a user interface for the visual commentary of yoga positions. This process heavily relied on research with potential users and involved collecting sets of annotated photographs on paper. From these, interesting and repeated elements were applied to the developed interface. Qualitative tests were repeatedly conducted throughout the process; I identified shortcomings from these observations, applied changes, and tested again. The result of the work is a demonstrative user interface implemented as a progressive web application.

Klíčové slová

yoga, športové pozície, komentovanie fotky, užívateľské rozhranie, UI, UX, návrh, Figma, React, webová aplikácia, progresívna webová aplikácia, prototypovanie

Keywords

yoga, sports positions, commenting on a photo, user interface, UI, UX, design, Figma, React, web application, progressive web application, prototyping

Citácia

BARBORÍK, Martin. *Výskum a vývoj UI/UX pre komentovanie športových pozícií*. Brno, 2024. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Adam Herout, Ph.D.

Výskum a vývoj UI/UX pre komentovanie športových pozícií

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána prof. Ing. Adama Herouta Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Martin Barborík
9. mája 2024

Podakovanie

Týmto by som sa chcel poďakovať vedúcemu mojej bakalárskej práce, pánovi prof. Ing. Adamovi Heroutovi Ph.D. Za rady na konzultáciách, testovanie mojich prototypov a poskytnutie témy tejto práce.

Obsah

1	Úvod	3
2	Joga, komentovanie pozícií	4
2.1	Výskum preferencií potenciálnych užívateľov	4
2.2	Zistené požiadavky na rozhranie	6
3	Princípy a pravidlá návrhu a vývoja moderných aplikácií	10
3.1	Don't make me think!	10
3.2	Modely správania	12
3.3	Princípy vizuálneho dizajnu	13
3.4	Návrhové vzory	13
3.5	Moderné technológie pre vytváranie prototypov: Figma	14
3.6	Zvažované Frameworky pre vývoj	16
3.7	React a zvolené knižnice	16
4	Výskum existujúcich aplikácií	21
4.1	Reddit	21
4.2	MIUI editor fotiek	21
4.3	Canva	22
4.4	Instagram	24
4.5	Picmarker	26
4.6	CoachNow	27
5	Návrh užívateľského rozhrania: prototypy a iteratívne vylepšovanie	29
5.1	Spôsob kreslenia	29
5.2	Panel s nástrojmi	30
5.3	Výber farieb	32
5.4	Nastavenie hrúbky tvarov	33
5.5	Transformátor	33
5.6	Mazanie objektov	34
5.7	Textový komentár	35
5.8	Zvýraznenie nakreslených tvarov	36
5.9	Zameranie pohľadu	37
5.10	Rozloženie	38
6	Implementácia a zhodnotenie	40
6.1	Implementácia	40
6.2	Zhodnotenie	45

7 Záver	48
Literatúra	49

Kapitola 1

Úvod

Táto bakalárska práca sa zaoberá výskumom a iteratívnym vývojom užívateľského rozhrania so zameraním na vizuálne komentovanie pozícií v joge (viac v kapitole 2). Rozhranie umožňuje inštruktorom a osobným trénerom efektívne napravovať a komentovať športové pozície pre zlepšenie formy cvičenca bez osobného kontaktu, formou fotografií. Cieľom je vytvoriť intuitívne užívateľské rozhranie, ktoré je prístupné pre potenciálnych užívateľov a vychádza z výskumu potrieb a preferencií tejto skupiny, ďalej popísaného v sekcii 2.1.

Pri návrhu a testovaní som vychádzal z princípov UI (užívateľské rozhranie) a UX (užívateľská skúsenosť) popísaných v kapitole 3, v tejto kapitole sa venujem okrem toho aj nástrojom a technológiám, s ktorými som sa oboznámil a použil počas vývoja. V kapitole 4 som vykonal výskum existujúcich aplikácií, ktoré som testoval alebo som sa s nimi inšpiroval pri riešení požiadaviek na rozhranie.

V kapitole 5 popisujem a zobrazujem ako sa upravovali časti rozhrania na základe testov. Veľa aspektov rozhrania vyžaduje interakciu, ktorá v prototypovacích nástrojoch nie je možná, preto som implementoval rozhranie ako progresívnu webovú aplikáciu vyvinutú v JavaScriptovej knižnici React, detaily implementácie sú v kapitole 6. Primárne zameranie je na zariadenia, ktoré podporujú aj gestá dotykcom. Rozhodol som sa pre formu progresívnej webovej aplikácie pre podporu zobrazenia a použitia podobnému natívnej aplikácii a jednoduchú inštaláciu na mobilných zariadeniach alebo počítačoch bez závislosti na operačnom systéme. Toto riešenie dostatočne simuluje funkčnosť a vzhľad natívnej aplikácie pre potreby tohto riešenia. Voľba webovej aplikácie uľahčuje taktiež testovanie na diaľku a podporuje zobrazenie okomentovanej pozície v prehliadači, bez potreby stiahnutia aplikácie. Aplikácia je dostupná na [Poses](#).

Kapitola 2

Joga, komentovanie pozícií

Joga je staroveká disciplína, pôvodom z Indie, ktorá sa zameriava na telo, myseľ a ducha. Z tradičného pohľadu ide o poslanie duše. Celkovo sa joga skladá z ôsmich končatín alebo stupňov: *yama* (morálne prikázania), *niyama* (sebačistenie disciplínou), *asana* (pozícia), *pranayama* (ovládanie dychu), *pratyahara* (zameranie zmyslov na vnútro), *dharana* (sústredenie na jeden bod), *dhyana* (meditácia), *samadhi* (zlúčenie meditujúceho s objektom meditácie) [5]. Moderná joga, teda joga, ktorú si predstavíme u nás, sa zameriava prevažne na *pranayamu*, *asanu*, *dhyanu*. V dnešnej dobe existuje veľa druhov jogy, ale pri každej je jedna vec konzistentná, cvičiaci sa musí naučiť ako robiť pozície.

Súčasťou mnohých pozícií je aj zameranie pohľadu známe ako *drishti* [12]. Ide o techniku, pri ktorej cvičiaci zameria pohľad na špecifické body, pre zachovanie pozornosti a vedomia o svojom tele. Oči by sa mali zavrieť až pri dosiahnutí perfektnej pozície [5].

2.1 Výskum preferencií potenciálnych užívateľov

Vytvorenie návrhu aplikácie vyžaduje porozumenie potenciálnym užívateľom. Táto kapitola sa venuje spôsobom akými som zisťoval potreby pre užívateľské rozhranie v neskorších kapitolách. Spôsobov ako vykonať výskum s užívateľmi je veľa, medzi zvažované patrili: interview, prieskumy cez dotazníky alebo úlohou na papieri.

2.1.1 Výskum s inštruktormi papierovou formou

Programy na kreslenie sú často inšpirované reálnym aktom kreslenia, nech už ide o použité nástroje, tlak pri kreslení atď. S touto myšlienkou som vykonával aj časť svojho výskumu. Vytvoril som sadu 20 fotiek s rôznymi jogovými pozíciami, na ktorých cvičiaci nemajú optimálnu formu. Každú fotku som vytlačil na papier A4 s dostatočným bielym priestorom pod fotkou. Osobám zúčastneným v tejto forme výskumu bolo poskytnuté nasledujúce:

- Sada 20 papierov s fotkou.
- Modré pero.
- Päť fixiek rôznych farieb (čierna, strieborná, zlatá, zelená, červená)

Pokyny pre inštruktora boli, na každej strane poskytnúť pomoc cvičiacej osobe s účelom zlepšenia formy pri budúcom cviku. Celkovo sa tejto časti výskumu zúčastnilo 18 inštruktorov. Komentovanie jogovej pozície na papieri poskytlo náhľad, akou formou by užívateľ

komunikoval rady v aplikácii. Výsledné papiere mali rozdielne prístupy od komentovania iba textom, kombinovanie textu a kresieb, až plne grafické riešenie úlohy. Príklady riešení, ktoré vykonali inštruktori sú viditeľné v obrázkoch 2.1, 2.2 a 2.3.

2.1.2 Rozhovory s inštruktormi

Ako druhú formu výskumu s potenciálnymi užívateľmi som zvolil rozhovor (interview). K tomuto rozhodnutiu som prišiel z dôvodu využitia dodatočných otázok k odpovediam inštruktora. Otázky boli zamerané na priblíženie problematiky, dávania spätnej väzby a komunikácie medzi inštruktormi a cvičiacimi. Táto forma menej zaťažila inštruktora, ktorí sa zúčastnili vyplnenia predošlej papierovej úlohy. Otázky rozhovoru sa zameriavali na nasledujúce časti:

Názor na nápad aplikácie

Kontaktovaní inštruktori mali na nápad digitálneho nástroja na komentovanie pozícií zmiešané pocity. Hlavné viditeľné rozdiely boli v aspektoch v akom cvičia a učia jogu. Ako bolo v kapitole 2 popísané, tak joga má aj svoju tradičnú duchovne založenú stránku, ktorá sa aplikuje aj v cvikoch, zatiaľ čo by sa aplikácia zameriavala príliš na vizuálny aspekt.

Taktiež je v dnešnej dobe rozdielny prístup k optimálnej pozícii a pozícii prispôbenej individuálnej osobe. Prístup k joge je rôzny. Niektorí inštruktori už ani nedávajú rady počas hodiny alebo iba vo veľmi zriedkavých prípadoch. Naopak sú inštruktori, ktorí dávajú aktívne rady počas hodín a boli by ochotní aj cez internet.

Zmysel v aplikácii vidia pre úroveň cvičiaceho medzi úplným začiatočníkom a mierne pokročilým (mierne pokročilý jogin by už mal byť samostatný).

Dôležitá bola u všetkých pripomienka zdravia. Osobe, s ktorou nemali osobný rozhovor, a nevedia zdravotné limity danej osoby, by nedali rady.

Záujmy inštruktora

Počas rozhovorov som zisťoval aj aké majú inštruktori záujmy, hlavne tie súvisiace s jogou alebo so svojimi osobnými zariadeniami. Odpovede boli rôzne. Najčastejšie zdieľané záujmy boli podobné športy ako napr. pilates, organizovanie výletov spojených s jogou, workshopy, knihy, zdieľanie príspevkov atď. Mnohí používajú sociálne siete typu Facebook, Instagram a WhatsApp.

Komunikácia s cvičencami

Inštruktori nemali jednu preferovanú platformu, ktorou by komunikovali. Inštruktori, ktorí so svojimi cvičencami komunikujú aj mimo hodín v štúdiu, často odpovedali, že pokiaľ ide o sociálne siete, ktoré používajú, nechávajú iniciatívu na cvičencoch. Medzi často používané patria Facebook, Instagram a WhatsApp, prípadne aj e-mail. Po Covide viacerí začali natáčať cviky naživo na platformách ako Facebook, Instagym alebo vlastné platformy štúdií.

Mobilné zariadenie alebo počítač

Pre komunikáciu a veľakrát aj nahrávanie cvikov preferujú mobilné zariadenia. Aplikácia na komentovanie teda môže využiť aj funkcionality dostupné len na dotykových zariadeniach. Našlo sa aj niekoľko inštruktora, ktorí preferovali notebook.

Využitie jogových aplikácií

Okrem aplikácií na nahrávanie cvikov a spravovanie štúdia, neboli u inštruktorov veľmi používané. Počuli od svojich cvičencov o aplikáciach na tvorenie sád cvikov.

2.2 Zistené požiadavky na rozhranie

Výsledky z prieskumu s inštruktormi a zistenia z papierovej úlohy poskytli dôležité poznatky pre formuláciu požiadaviek na návrh aplikácie na komentovanie jogových pozícií. Tieto požiadavky zohľadňujú potreby reálnych používateľov a sú kľúčové pre návrh funkčného a intuitívneho užívateľského rozhrania.

- **Intuitívnosť a jednoduchosť:** Rozhranie by malo byť intuitívne a jednoduché na používanie, aby bolo prístupné aj pre menej technicky zdatných užívateľov.
- **Úroveň v joge:** Rozhranie by malo brať do úvahy, že proces komentovania vykonáva užívateľ, ktorý je pokročilý v daných cvikoch a teórii, zatiaľ, čo prijímať okomentované fotografie bude osoba bližšie k začiatkovej úrovni.
- **Grafická spätná väzba:** Rozhranie by malo poskytovať vizuálnu spätnú väzbu na jogové pozície, pomocou možnosti kreslenia na fotografiách.
- **Textová spätná väzba:** Rozhranie by malo poskytovať textový vstup pre doplnenie detailov alebo pridanie odporúčení.
- **Využitie farieb:** Rozhranie by malo poskytovať niekoľko farieb v palete pre odlíšenie častí pri komentovaní.
- **Obmedzenie sťahovania:** Rozhranie by mohlo obmedziť potrebu naplnenia galérie užívateľa.
- **Zdravotné problémy:** Rozhranie by nemalo byť zamerané na komentovanie neznámych osôb.
- **Zdieľanie:** Potenciálni užívatelia využívajú rôzne platformy na komunikáciu s cvičencami. Malo by podporovať zdieľanie do iných aplikácií alebo byť súčasťou vlastnej sociálnej siete.
- **Zariadenie:** Rozhranie by malo byť primárne zamerané na mobilné rozhranie, ale podpora počítača by bola pozitívna.
- **Farby:** Rozhranie je zamerané na značenie do fotky. Preto sú kontrastné farby dôležité, ale zároveň treba brať do úvahy povahu osôb praktikujuúcich jogu. V joge ide o uvoľnenie a meditáciu. Farby by mali byť pokojnejšie. Ak už nie pre nástroje, tak aspoň prvky ako tlačidlá a pozadie.

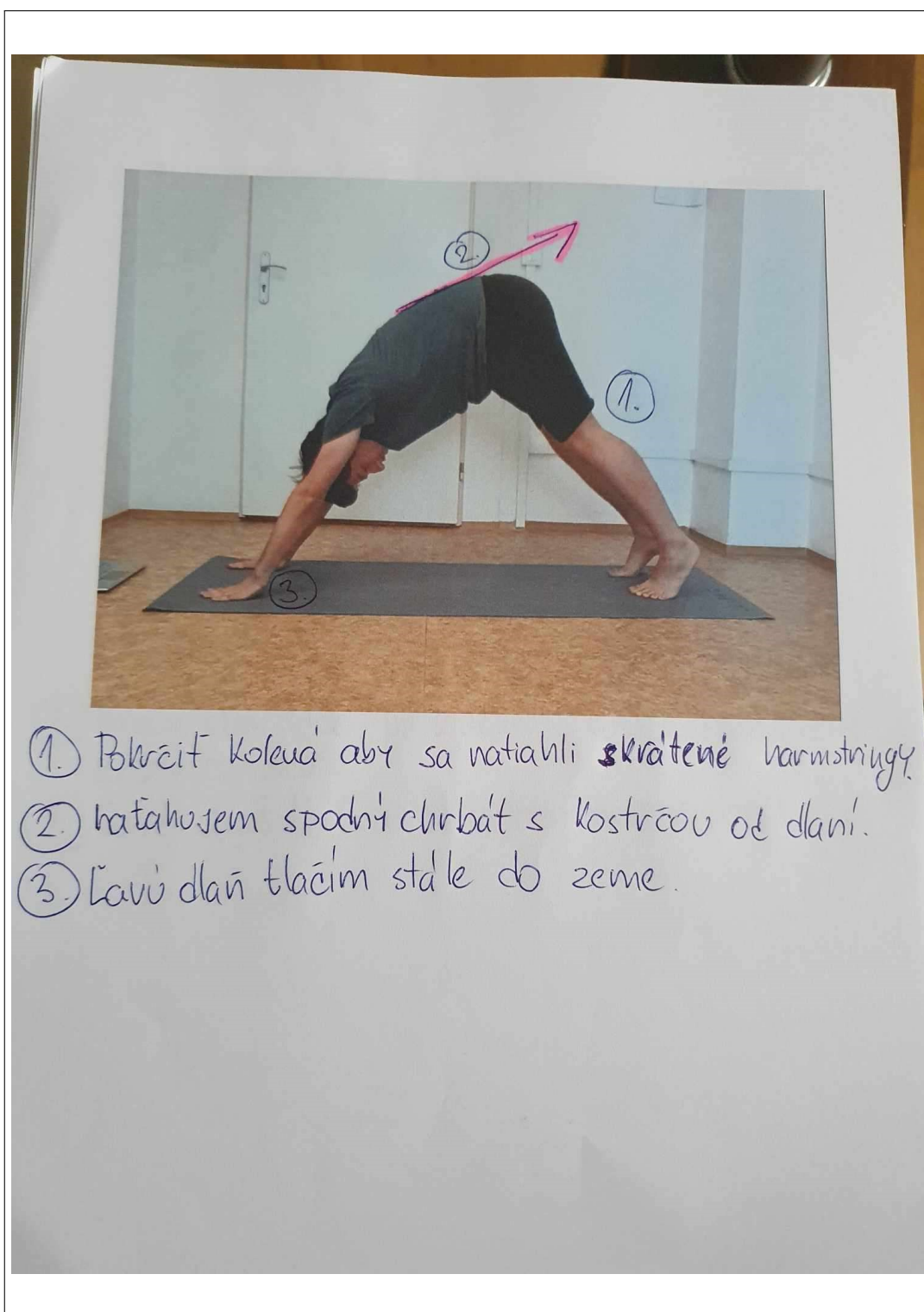


Obr. 2.1: Príklad riešenia úlohy na papieri s fixkami bez použitia textového popisu. Inštruktor napriek viacerým poskytnutým farbám využil jednoduché tvary jednej farby.



HRUDNÍK trochu mác „poutlačit“ → aby boli
pazě a chrbtica v jednej linii.
Päty postupne pri lepšej flexibilitě ťáhať až na zem

Obr. 2.2: Príklad riešenia úlohy na papieri s fixkami použitím grafického aj textového popisu. Inštruktor využil na odlíšenie rôznych častí odlišné farby a farbu zakomponoval aj do textového popisu pod obrázkom.



Obr. 2.3: Príklad riešenia úlohy na papieri s použitím značiek a textu. Účastník vo väčšine pozícií popisoval rady textom, prípadne použil minimálne grafické prvky ako šípky. Jedna z hlavných inšpirácií zakomponovania textu v rozhraní tejto práce.

Kapitola 3

Princípy a pravidlá návrhu a vývoja moderných aplikácií

V tejto kapitole popisujem hlavné pravidlá a princípy, s ktorými som sa oboznámil v knihách a článkoch. Venujem sa použiteľnosti a jej testovaniu, návrhovým vzorom, nástrojom na prototypovanie a frameworkom pre vývoj.

3.1 Don't make me think!

Hlavné pravidlo pri vytváraní použiteľnej aplikácie je podľa Steva Kruga [9]: „Don't make me think“. Nie maximálny počet kliknutí alebo konzistencia. Samozrejme, aj tieto pravidlá pomôžu. Prvoradé je, aby užívateľ pri pohľade na aplikáciu pochopil jej účel a ako ju môže použiť. V prípade komplexnejších aplikácií alebo aplikácií s novým konceptom pre užívateľa je trochu premýšľania očakávané, ale úsilie by malo byť malé. Cieľom tejto práce je vytvoriť aplikáciu použiteľnú pre cieľového užívateľa. Ako je možné aplikáciu označiť ako použiteľnú? Steve Krug vo svojej knihe [9] definoval použiteľnosť pomocou týchto bodov:

- **Užitočná** – Aplikácia poskytuje užívateľom niečo, čo potrebujú urobiť.
- **Naučiteľná** – Užívateľ používaním príde na funkcionality aplikácie.
- **Zapamätateľná** – Užívateľ sa nemusí pri každom otvorení učiť použitie nanovo.
- **Efektívna** – Užívateľ dokáže splniť úlohu, na ktorú je aplikácia určená a zároveň v rozumnom čase, bez priveľkého úsilia.
- **Žiadúca** – Existujú potenciálni užívatelia, ktorí by mali o použitie aplikácie záujem.
- **Zaujímavá** – Aplikácia je zábavná na použitie.

3.1.1 Testovanie použiteľnosti

K zaručeniu použiteľnosti aplikácie neexistuje lepší spôsob ako pozorovať potenciálneho užívateľa. Skúšať aplikáciu a písať si poznámky, ak dôjde k problému alebo nepochopeniu. Samozrejme testovať, až keď je aplikácia plne funkčná je príliš neskoro. S testovaním treba začať ihneď a robiť ho počas celého priebehu tvorby návrhu a implementácie.

Steve Krug vo svojej knihe *Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems* [8] popísal druh testovania, ktorým som sa riadil

aj ja pri vývoji, „Do-It-Yourself Usability testing“. Ide o kvalitatívny spôsob testovania, kde na rozdiel od kvantitatívneho, nejde o dokazovanie pomocou štatistík úspešnosti, časov splnenia úloh a podobných metrík, ale skôr o získanie pohľadu do premýšľania užívateľov, nájdenie nedostatkov a vylepšenie aplikácie z pozorovania a počúvania užívateľa. Dôležité poznatky z tejto knihy sú popísané v nasledujúcich podkapitolách.

Ako často testovať

Podľa Kruga je testovanie dôležité počas celého procesu vývoja. Odporúča testovanie formou „Jedno ráno za mesiac“. Počas ktorého otestuje aplikáciu s tromi užívateľmi. Tento prístup považuje za dostačujúci, odhalí sa pri ňom dostatok nedostatkov na opravu do ďalšieho testovania. Jedno testovanie za mesiac nepovažuje za maximum, ale minimum. Pokiaľ je potreba testovania, tak je častejšie testovanie vhodné.

Koho testovať

Výber osôb na testovanie môže byť komplikovaný. Je vhodné vyberať len ľudí, ktorí spadajú do kategórie cieľových užívateľov alebo môžeme testovať aplikáciu na osobách mimo? Steve Krug odporúča testovať aj mimo cieľovej skupiny. Nie všetky funkcionality aplikácie vyžadujú odbornosť cieľovej skupiny. Primárne na začiatku vývoja nemusia byť všetci odborníci. Bežné nedostatky sa nájdu pri bežných osobách. Dôležité je sa zamyslieť, či by mal aj cieľový užívateľ daný problém. Ďalší dôvod prečo je vhodné testovať aj na ostatných je brať do úvahy ľudí, ktorí nemusia plne poznať terminológiu experta, pretože ešte nie sú na tej úrovni.

Jedno kolo testovania by malo mať troch účastníkov. Hlavným cieľom „Do-it-yourself“ testovania je odhaliť hlavné problémy návrhu. S väčším počtom účastníkov sa hľadajú detaily. Je lepšie radšej vykonať viac kôl testovaní.

Čo testovať

Testovať finálny produkt nie je správny prístup. Testovanie by malo začať už v prvých fázach vývoja. Odporúčanie od Kruga je začať skôr ako to dáva zmysel. Nezáleží na tom, že nie je produkt plne funkčný. Čo sa má teda testovať?

- **Existujúci produkt** – V prípade, že už máme hotovú aplikáciu, je výber toho, čo testovať, jednoduchý. Treba začať tu. Nájsť nedostatky a poučiť sa z nich.
- **Aplikácie ostatných** – Predtým, než začne vývoj našej aplikácie, môžeme sa pozrieť na existujúce riešenia, ktoré riešia náš alebo podobný problém a otestovať ich s užívateľmi.
- **Náčrty** – Na začiatku plánovania je bežné dávať všetky nápady na papier a robiť si náčrty. Nie je ich možné testovať dôkladne. Ide o rýchle testovanie formou popisu toho, čo osoba vidí.
- **Wireframe** – Ďalšia forma rýchleho testovania zameraného na kategorizovanie, pomenovanie prvkov a navigáciu.
- **Prototypy a implementácie** – Hlavná forma testovania počas vývoja, kde sa testuje interakcia s rozhraním.

Priebeh testovania

Počas prípravy na testovanie je potrebné pripraviť úlohy pre užívateľov. Zamerať sa na veci, ktoré chceme otestovať a vytvoriť z nich scenár. Samotný priebeh testovania prebieha nasledovne:

- **Uvítanie** – Na začiatku sa účastníkovi vysvetlí testovanie, pripomenie sa mu, že sa testuje aplikácia a nie on.
- **Otázky** – Pokladanie otázok o účastníkovi, zistenie ich zručností týkajúcich sa technológií, preferencií a podobne.
- **Popis stránky** – Účastník si obzerá stránku, popisuje čo vidí. Premýšľa nahlas.
- **Úlohy** – Účastník vykonáva vopred určené úlohy alebo jednu dlhšiu úlohu. Je dôležité, aby účastník hovoril nad čím premýšľa pri používaní aplikácie. Pokladajú sa mu iba otvorené otázky a nedávajú sa žiadne rady.
- **Dodatočné otázky** – Pokladanie otázok účastníkovi vytvorených počas plnenia úloh.
- **Ukončenie** – Poďakovanie a rozlúčenie. Nasledovne sa spíše zoznam všetkých problémov a určia sa tri najdôležitejšie.

Testovanie na diaľku

V niektorých prípadoch je jednoduchšie vykonať test cez internet. Priebeh testovania na diaľku je podobný osobnému. Rozdiel je v poskytnutí testovanej aplikácie účastníkovi a potreba aplikácie na zdieľanie pohľadu obrazovky.

3.2 Modely správania

Porozumenie tomu ako sa užívateľ správa pri používaní aplikácie je dôležité pre vývoj. Hlavné koncepty, s ktorými som sa oboznámil sú popísané v nasledujúcich podkapitolách.

3.2.1 Satisficing

Satisficing [6] je termín, ktorý spája dva koncepty *satisfying* (uspokojivý) a *sufficing* (postačujúci). Je to spôsob akým ľudia pristupujú k vykonávaniu úloh. Mnohí nehľadajú najlepší spôsob ako niečo vykonať, ale skôr prvé riešenie, ktoré úlohu splní dostatočne.

3.2.2 Habituácia

Užívatelia si používaním aplikácií zvykajú na akcie, ktoré vykonávajú a časom nad nimi nemusia premýšľať [6]. Zároveň si tieto zvyklosti prenášajú medzi podobnými aplikáciami. Príkladom na zariadeniach s klávesnicou sú skratky:

- Ctrl-Z – vrátenie kroku
- Ctrl-Y – vykonať znovu

Pri návrhu aplikácie pre vizuálne komentovanie na fotke je dôležité testovať aké gestá užívateľa od aplikácie očakávajú, dodržiavaním zvyklostí sa dá predísť problémom, na ktoré by užívatelia mohli naraziť. Počas testov väčšina inštruktorov skúšala využiť gestá s viacerými dotykmi, ktoré sa naučili z aplikácií na zobrazenie fotiek a úpravu na sociálnych sieťach.

3.2.3 Priestorová pamäť

Užívatelia si častejšie pamätajú prvky obrazovky pomocou lokality než podľa názvu. Príkladmi sú napríklad umiestnenia tlačidiel na potvrdenie akcie a na zrušenie. V aplikáciach na kreslenie hľadajú nástroje v pozícií relatívnej k ostatným nástrojom, pozerú sa preto do panelu s nástrojmi. Preto dynamické zmeny v menu alebo v paneloch môžu viesť k problémom. Tento jav sa volá priestorová pamäť [6].

3.3 Princípy vizuálneho dizajnu

Kniha UX for Beginners [10] približuje ako vizuálny vzhľad ovplyvňuje premýšľanie užívateľov. Z popísaných som vybral tie, ktoré súvisia s mojou prácou.

3.3.1 Farba

V UX má farba iný význam ako v UI, nejde len o vzhľad, ale aj o funkcionality. Každá farba má v kontexte svoj vlastný význam. Napríklad na stránke na nakupovanie je červená na zrušenie objednávky, modrá na potvrdenie a neutrálna farba reprezentuje zrušenie akcie.

3.3.2 Proximita

Proximita je blízkosť objektu k ďalším objektom na obrazovke. Objekty, ktoré sú blízko pri sebe vyvolávajú u užívateľa dojem, že patria k sebe.

3.3.3 Opakovanie

V prípade, že za seba dáme niekoľko rovnakých elementov rozhrania, tak užívateľ nebude vidieť osobitné elementy, ale spojí si ich do tvaru, uvidí čiaru. V prípade, že použijeme trochu iný element alebo ho od ostatných oddialime, tak naň upútame pozornosť. Tento jav sa volá porušenie vzoru.

3.4 Návrhové vzory

V oblasti návrhu užívateľského rozhrania sa často riešia rovnaké problémy, ktoré vyriešili už existujúce rozhrania. Z tohto dôvodu vznikli zaužívané metódy, ktoré riešia špecifický problém. V kontexte komentovania pozícií na fotkách sú to prehliadač médií a plátno s paletou rozoberané v nasledujúcich sekciách.

3.4.1 Prehliadač médií

Návrhový vzor pre prehliadanie a výber z množstva objektov [6]. Štruktúra tohto vzoru je mriežka, kde sú objekty usporiadané do riadkov a stĺpcov. V kontexte grafickej aplikácie poskytuje pre každý objekt obrázok a pri potrebe aj metadáta. Ide o veľmi bežný vzor pre prehľadné zobrazovanie médií.

3.4.2 Plátno a paleta

Návrhový vzor, ktorý sa používa pri práci s obrázkami s účelom značenia, podobnému papierovej forme z kapitoly 2.1, je plátno a paleta [6]. Plátno predstavuje veľké prostredie obrazovky, kde sa vykonáva kreslenie a upravovanie. Paleta tvorí malú časť obsahujúcu nástroje a nastavenia nástrojov reprezentované ikonkami. Spôsobov ako užívateľ využíva nástroje je veľa, vytváranie tvarov na plátne je možné cez ťahanie od ikonky na plátno, označením nástroja a kliknutím, kliknutím a ťahaním atď. Bližšie sa tieto spôsoby popisujú v kapitole 5.1 s návrhom.

3.5 Moderné technológie pre vytváranie prototypov: Figma

Figma je webová a desktopová aplikácia na návrh a prototypovanie webových stránok a aplikácií. Figma umožňuje spoluprácu na projekte v reálnom čase. Tento nástroj je možné využiť na viacero vecí, umožňuje tvorenie poznámok a náčrtov na začiatku výskumu, neskôr tvorenie wireframov a prototypov na testovanie s potenciálnymi užívateľmi. Dôležité prvky o Figue som čerpal z knihy „Designing and Prototyping Interfaces with Figma“ [13].

3.5.1 Rámce a skupiny

Základnými stavebnými prvkami v prostredí Figma sú rámce (frames) a skupiny (groups). Rámce môžu byť vložené jeden do druhého a obsahujú možnosti ako auto-layout, mriežky pre rozloženie a všetky funkcie potrebné pre prototypovanie. Na rámce sa dá pozeráť ako na kontajnery, do ktorých sa vkladajú ďalšie prvky, aj ďalšie rámce. Figma obsahuje preddefinované rámce rôznych rozmerov podľa známych zariadení.

Skupiny vo Figue slúžia na organizáciu objektov, ktoré logicky spolu súvisia. Umožňujú manipulovať s viacerými objektami naraz a aplikovať zmeny na celú skupinu, bez potreby meniť každý objekt individuálne. Skupiny je možné použiť aj pre organizáciu, na zachovanie prehľadnosti v štruktúre rozhrania.

3.5.2 Komponenty a varianty

Komponenty sú opakovane použiteľné prvky, ktoré sa definujú raz a potom sa môžu používať na viacerých miestach v projekte. Zmeny na komponente sa automaticky prejavajú na všetkých jeho inštanciách. Toto je obzvlášť užitočné napríklad pri práci s tlačidlami, ikonami alebo inými elementmi rozhrania, ktoré majú byť konzistentné v celom projekte.

Varianty sú rozšírením konceptu komponentov. Umožňujú vytvárať rôzne verzie komponentu pod jedným spoločným názvom. Komponenty, ktoré sa líšia len farbami nemusia byť ako samostatné komponenty, ale môžu byť varianty jedného. Napríklad tlačidlá na potvrdenie, zrušenie a zmazanie sa líšia farbou a textom, môžu byť variantami tlačidla alebo rôzne stavy tlačidla, neutrálny stav, stlačený atď.

3.5.3 Prototypovanie

Prototypovanie vo Figue umožňuje poskytnúť interakciu k statickému návrhu a tým vytvoriť prototyp na testovanie s užívateľmi, kde môžu klikáť na komponenty a sledovať ako sa mení pohľad.

Prechody a spúšťače

Figma poskytuje rôzne typy prechodov ako napríklad Move, Push a Slide, ktoré pomáhajú simulovať správanie reálnej aplikácie.

Triggers (Spúšťače):

- **On Click/On Tap:** Predvolený spúšťač, ktorý reaguje na kliknutia.
- **On Drag:** Reakcia na kliknutie a ťahanie.
- **While Hovering:** Aktivuje sa, keď myš prejde nad konkrétnu oblasť.
- **While Pressing:** Reaguje na dlhé stlačenia alebo kliknutia a držanie.
- **Key/Gamepad:** Spúšťa akcie na základe klávesov na klávesnici alebo vstupov gamepadu.
- **Mouse Enter/Mouse Leave:** Reakcia pri vstupe alebo výstupe z určitej oblasti .
- **Mouse Down/Mouse Up:** Spustenie začiatkom alebo koncom kliknutia myšou, prípadne dotykcom na obrazovke.
- **After Delay:** Vykonalenie akcie po nastavenom oneskorení.

3.5.4 Animácia so Smart Animate

Smart Animate umožňuje ľahko animovať prechody medzi dvoma stavmi. Identifikovaním zhodných vrstiev medzi rámcami Figma automaticky vytvára animácie podľa zmeny v polohe, veľkosti alebo vzhľade prvkov. S touto funkciou je možné vytvoriť komponenty ako slider jednoducho.

3.5.5 Interaktívne komponenty

Interaktívne komponenty zlepšujú interaktivitu prototypov tým, že umožňujú definovať interakcie v komponente, ktoré sa automaticky použijú aj na inštanciách daného komponentu. Tento spôsob znižuje počet zbytočných pohľadov.

3.5.6 Premenné

Premenné vo Figma umožňujú opätovné použitie údajov v celom návrhu, čo uľahčuje udržanie konzistencie pre štylizovanie alebo rozšíriť interakciu. Môžu uchovávať hodnoty, ako sú farby, čísla alebo textové reťazce, ktoré je potom možné použiť v rôznych častiach návrhu. Vďaka premenným som mohol nastavovať farbu ostatných komponentov priamo zo svojho komponentu pre paletu farieb, bez potreby vytvorenia pohľadu pre každú zvolenú farbu.

3.5.7 Testovanie

Po dokončení prototypu je nutné navrhnuté prvky otestovať, na tento účel je možné použiť prezentačný mód priamo vo webovej/desktopovej aplikácii. Účastník testovania môže použiť toto zariadenie alebo mu môže byť poslaný odkaz, pomocou ktorého si vie otvoriť prototyp u seba. Druhá možnosť je použiť mobilnú aplikáciu Figma – prototype mirror share, pomocou ktorej je možné prejsť prototyp na mobilnom zariadení v natívnej aplikácii.

3.6 Zvažované Frameworky pre vývoj

V roku 2024 je výber správneho frameworku pre vývoj aplikácií ťažkým rozhodnutím, vzhľadom na široké spektrum dostupných a populárnych technológií. V tejto sekcii poskytnem prehľad troch populárnych frameworkov. Moje výbery ovplyvnil fakt, že na vytvorenie demonštračného rozhrania nepotrebujem fullstack framework.

3.6.1 Svelte

Svelte [4] je javascriptový framework určený na vývoj webových aplikácií. Svelte je výnimočný tým, že využíva kompilátor, transformuje kód napísaný vo Svelte do JavaScriptu, čím zabezpečuje rýchlejší výkon. Jednou z hlavných výhod Svelte je jeho prístup k manipulácii s DOM (Document Object Model). Na rozdiel od iných populárnych frameworkov, Svelte nepoužíva virtuálny DOM. Miesto toho priamo manipuluje s reálnym DOM, čo vedie k rýchlejšiemu vykreslovaniu a zlepšenému výkonu. Zároveň je jeho kód ľahko čitateľný, pretože je podobný klasickému HTML a JavaScriptu. Nevyžaduje žiadne volania funkcií na uchovanie stavov, stačí použiť `let example = 1`.

SvelteKit, je nadstavbou Svelte, poskytuje SSR (server-side rendering), automatické delenie kódu, súborovo orientované smerovanie a ďalšie.

3.6.2 React

React [14] je open-source JavaScriptová knižnica, široko používaná na vytváranie užívateľských rozhraní pre webové aplikácie. React prináša svoju efektivitu vývoja prostredníctvom optimalizovaného virtuálneho DOM, ktorý zabezpečuje zmeny v DOM iba v prípade zmien. React podporuje deklaratívny prístup k programovaniu a znovupoužitie komponentov, čo znižuje redundanciu kódu a zvyšuje údržbu a konzistenciu aplikácií. Hlavná výhoda Reactu je však veľká komunita a podpora.

3.6.3 React Native

React Native [14] je framework na vývoj mobilných aplikácií na Android alebo IOS. Používa JavaScript na vývoj mobilných rozhraní s podobným prístupom ako React. Potrebu jazykov, ktoré sa používajú na vývoj natívnych aplikácií ako Java nevyžaduje, ale podporuje. Napríklad na hlbšie optimalizovanie.

3.7 React a zvolené knižnice

Po zvažovaní viacerých frameworkov a diskusií s vedúcim práce som sa rozhodol pre implementáciu aplikácie v knižnici React. React má veľkú komunitu a umožňuje lepšie prispôbenie. S knižnicou už mám aj skúsenosti, preto za účelom tvorby užívateľského rozhrania na demonštrovanie výsledkov a priebežné testovanie prvkov, ktoré nie je možné vykonať vo Figma (kapitola 3.5) považujem React za dobrú možnosť. Zároveň je možné aplikáciu napísanú v Reacte jednoducho previesť na progresívnu webovú aplikáciu (PWA) a testovať rozhranie na mobile. Základný popis knižnice je v kapitole 3.6, v tejto kapitole ho však vysvetlím podrobnejšie. Verzia knižnice, v ktorej som vyvíjal rozhranie je 18.2.0. Od 16.8.0. sa v Reacte prešlo od komponentov vytváraných pomocou tried na funkcionálne komponenty a udržiavanie hodnôt v komponente sa vykonáva pomocou hooks. V tejto sekcii vysvetlím

základné koncepty Reactu a popíšem použité knižnice pre tvorbu implementácie v kapitole 6.1.

3.7.1 Základné koncepty Reactu

Základné koncepty Reactu, ktoré vychádzajú z oficiálnej dokumentácie [11], sú:

Komponenty

Komponenty sú základné stavebné prvky Reactu. Sú to nezávislé, znovupoužiteľné časti užívateľského rozhrania, ktoré môžu obsahovať vlastný stav a logiku.

State (stav)

Stav komponentu obsahuje hodnoty, ktoré ovplyvňujú, ako sa komponent vykresluje, zmeny v stavoch spôsobujú prekreslenie. Môže sa meniť pomocou set funkcií a je nevyhnutný pre dynamické rozhranie. So stavom by sa malo pracovať ako read-only. Pri zmene hodnôt by sa mala vytvoriť nová kópia.

Props

Props sú argumenty, ktoré sa predávajú komponentom. Pomocou props sa poskytujú údaje z nadradeného komponentu do podradených komponentov. Opačným smerom to nie je možné, preto sa stav a funkcie na nastavenie stavu vytvárajú niekedy v otcovských komponentoch a predávajú cez props synovským komponentom.

Hooks

Hooks sú funkcie, ktoré umožňujú React komponentom používať stav a ďalšie funkcie Reactu. Existuje niekoľko druhov:

- **State Hooks** – Umožňujú komponentom udržiavať hodnoty. Na pridanie stavu do komponentu sa môže použiť jeden z nasledujúcich Hooks:
 - **useState** – Deklaruje stavovú premennú, ktorú je možné priamo aktualizovať pomocou set funkcie.
 - **useReducer** – Deklaruje stavovú premennú s logikou aktualizácie vnútri funkcie reducer.
- **Context Hooks** – Umožňujú komponentu získať informácie zo vzdialených otcovských komponentov bez prenášania informácií cez props.
 - **useContext** – Číta a prihlási sa na kontext.
- **Ref Hooks** – Umožňujú komponentu udržiavať odkazy na DOM elementy alebo iné hodnoty, ktoré sa nepoužívajú na vykresľovanie, ale sú nevyhnutné pre rôzne operácie. Pri zmene nevyvolávajú prekreslenie.
 - **useRef** – Deklaruje referenciu ref, ktorá môže udržiavať ľubovoľnú hodnotu. Zmena nevyvoláva prekreslenie.

- **Effect Hooks** – Umožňujú komponentu sledovať externé prvky a synchronizovať sa pri ich zmene.
 - **useEffect** – Pripája komponent k externému systému.
- **Performance Hooks** – Pomáhajú optimalizovať výkon prekresľovania komponentov.
 - **useMemo** – Umožňuje uložiť hodnotu náročného výpočtu do vyrovnávacej pamäte.
 - **useCallback** – Umožňuje ukladať definíciu funkcie do vyrovnávacej pamäte medzi prekresľovaním komponentu.

3.7.2 Vite a Vite PWA plugin

Vite [3], je nástroj na urýchlenie tvorby projektov webových aplikácií. Poskytuje Hot Module Replacement (HMR) na takmer okamžité aktualizácie pri zmene v zdrojovom kóde. V dnešnej dobe je úplnou náhradou pre Create React App, ktorý bol dlhodobo využívaným nástrojom na zostavenie projektov v Reacte.

Predtým než sa dostanem k Vite PWA plugin, vysvetlím pomocou článku [7] progresívne webové aplikácie. Známe tiež ako PWA, sú to webové aplikácie, ktoré poskytujú pocit používania natívnej mobilnej aplikácie. Bežia pri tom v pozadí cez prehliadač. Na konvertovanie webovej aplikácie do PWA je treba vytvoriť `manifest.json`, je to súbor, ktorý poskytuje dáta o aplikácii. Môže to byť názov, farba, ikonky alebo iné. Následne je to **service worker**. Ide o súbor, ktorý sa zaregistruje v aplikácii a slúži ako proxy medzi danou aplikáciou a webom. Bežia v pozadí a vybavujú požiadavky, aj keď je aplikácia offline. Posledné je ukladanie do vyrovnávacej pamäte, slúži na rýchle fungovanie a funkciu aplikácie bez pripojenia na internet.

Vite PWA Plugin, je dodatočný modul k Vite, ktorý zabezpečuje fungovanie webovej aplikácie ako PWA. Stačí do `plugins` vo `vite.config.js` pridať `VitePWA` a zdefinovať `manifest`. Registrovanie skriptu `service worker` aj ukladanie do vyrovnávacej pamäte už vykoná `vite PWA plugin`.

3.7.3 Redux

Redux [1] je populárna knižnica pre správu stavu v aplikáciách napísaných v JavaScripte, často používaná s Reactom. Umožňuje sledovanie zmien vo vývojovom prostredí prehliadača. Pre lepšie porozumenie Reduxu je dôležité pochopiť základné koncepty ako Redux Toolkit, Store, Reducer, Slice a Dispatch.

Redux Toolkit

Redux Toolkit [1] je oficiálne odporúčaná sada nástrojov pre efektívny vývoj Redux aplikácií. Uľahčuje konfiguráciu store, písanie reducerov, vytváranie akcií a vykonávanie aktualizácií stavu. Cieľom Redux Toolkitu je zjednodušiť používanie Reduxu tým, že poskytuje predvolené nastavenia a odstraňuje potrebu viacerých balíčkov. Umožňuje písať menej kódu, uľahčuje prácu s Redux a zabezpečuje odporúčaný spôsob programovania v Redux prostredí.

Store

Store [1] je objekt, ktorý uchováva celý stav aplikácie. V Reduxe existuje len jeden globálny store pre celú aplikáciu, čo uľahčuje správu stavu. Obsah tohto objektu je poskytnutý aplikácii pomocou komponentu `<Provider store={store}/>`.

Reducer

Reducer [1] je čistá funkcia (to znamená, že nepracuje s vedľajšími účinkami), ktorá prijíma predchádzajúci stav aplikácie a akciu, ktorá bola vyvolaná, a na základe tejto akcie vráti nový stav. Základným princípom je, že reducer mení stav aplikácie tým, že neupravuje priamo predchádzajúci stav, ale vytvára a vracia nový objekt stavu.

Slice

Predstavuje časť stavu aplikácie spravovanú konkrétnym reducerom. Každý slice [1] zodpovedá za určitú doménu alebo funkčnú oblasť aplikácie a obsahuje vlastný stav a logiku na jeho spracovanie.

Dispatch

Dispatch [1] je metóda, ktorá sa používa na vykonanie akcie. Akcia je objekt, ktorý popisuje, čo sa má stať. Ide v podstate o vyvolanie udalosti s akciou, ktorú spracuje reducer.

3.7.4 Styled Components

Styled Components [2] je knižnica pre React, ktorá poskytuje písanie CSS štýlovania priamo do JSX súboru spolu s JavaScriptom. Takto vytvorené komponenty nevyžadujú mapovanie medzi komponentami a štýlmi, súčasne sa obchádza aj možný problém kolízie názvov tried, pretože komponenty vytvorené so Styled Components majú vždy unikátnu triedu. Ďalšou výhodou využitia Styled Components je práca s dynamickým štýlovaním pomocou props. Príklad využitia je útržok kódu z mojej implementácie tlačidla na výber farby (výpis 3.1).

```
const StyledColorButton = styled.button`
  display: block;
  aspect-ratio: 1;
  pointer-events: auto;
  margin: 0.1rem;
  border-radius: 50%;
  background-color: ${(props) => props.$color};
  border: none;
  width: ${(props) => (props.$isActive ? '1.7rem' : "1.6rem")};
  height: ${(props) => (props.$isActive ? '1.7rem' : "1.6rem")};
  cursor: pointer;
`;
```

Výpis 3.1: Styled Component pre tlačidlo na výber farby s dynamickým štýlom pomocou props.

3.7.5 Konva

Konva.js [15] je 2D grafická knižnica, ktorá umožňuje tvorbu a manipuláciu s grafickými prvkami, ako sú tvary, obrázky a text. Je obzvlášť užitočná pre vytváranie komplexných interaktívnych grafických užívateľských rozhraní.

Integrácia s Reactom – V mojej implementácii som použil React-Konva, ide o abstrakciu prvkov Konva do formy komponent Reactu, pre deklaratívny prístup k programovaniu, ktoré React ponúka. Použitie Konva v Reacte je popísana v oficiálnej dokumentácii¹.

Základné komponenty:

- **Stage** – Základný kontajner obsahujúci vrstvy a ich všetky tvary a skupiny.
- **Layer**: Vrstvy sa používajú na zoskupenie tvarov, čo umožňuje efektívnejšie prekresľovanie a špecifické operácie pre jednotlivé vrstvy. Oddelením prvkov do rôznych vrstiev možno obmedziť počet prekreslení. Prekresľujú sa prvky vo vrstvách, v ktorých došlo k zmene.
- **Tvary** – Konva ponúka veľké množstvo preddefinovaných tvarov ako sú kruhy, čiary, šípky a ďalšie, pričom každý tvar má svoje vlastnosti ako pozícia, veľkosť, farba výplne, okraj a mnoho ďalších, ktoré možno ľahko nastaviť.
- **Transformácie** Tvary na plátne môžu byť presúvané, otáčané atď. Knižnica poskytuje aj vlastný transformátor.
- **Group** – Skupiny umožňujú kombinovať viacero tvarov do jedného celku. Je to užitočné pri aplikovaní transformácií.
- **Canvas Context** – Umožňuje priamy prístup k plátnu pre komplexnejšie vykreslenie, ktoré nie je možné v Konva abstrakcii.
- **Drag and Drop** – Knižnica implementuje funkcionality drag and drop, pre jednoduché presúvanie prvkov na plátne.

¹<https://konvajs.org/docs/react/Shapes.html>

Kapitola 4

Výskum existujúcich aplikácií

Popri výskume s užívateľmi som hľadal aplikácie, ktoré riešia zistené potreby užívateľského rozhrania. Vyhľadával som najprv aplikácie, ktoré sa venujú cvičeniu jogy. Zistil som však, že dostupné aplikácie sa zameriavajú na vytváranie sád cvikov, sledovanie cvikov a spravovanie tréningu. Prehľadával som aj sociálne siete a fóra. Zameral som sa, ale hlavne na aplikácie na anotáciu fotiek. V nasledujúcich sekciách popisujem aplikácie, ktoré ma inšpirovali. Zameriavam sa primárne na spôsob vytvárania a úpravu tvarov a textu, mazanie a farebné palety. V tejto kapitole popisujem niekoľko aplikácií, ktoré ma inšpirovali pri mojom návrhu.

4.1 Reddit

Reddit je platforma online komunity, ktorá umožňuje používateľom vytvárať a diskutovať o témach v samostatných subredditoch, Subreddit [r/yoga](#)¹ je komunita zameraná na jogu. Členovia si vytvorili značenie *COMP* (Check Out My Pose) na príspevky, kde používatelia zdieľajú svoje fotografie, aby získali radu a spätnú väzbu, formou textových komentárov pod fotkou, od ostatných členov komunity.

4.2 MIUI editor fotiek

Prvé miesto, kde som hľadal nástroj na komentovanie fotiek bola galéria na mojom zariadení. Okrem rôznych funkcií na aplikovanie filtrov a pridanie nálepiek obsahuje tento editor kreslenie aj vkladanie textu. Vytvorené vektory je možné upravovať pomocou transformátora. Na rozdiel od neskoršie spomenutých aplikácií podporuje aj zväčšovanie a zmenšovanie označených objektov plátna pomocou multi-touch gest. Bez označenia objektu fungujú gestá na približovanie a oddialenie fotky. Hrúbka čiar sa nastavuje klikaním na komponent na začiatku palety s farbami. Toto riešenie je jednoduché a zaberá malý priestor na obrazovke. Textové komentáre sa vkladajú kliknutím v palete a umiestnia sa do stredu obrazu a opätovným kliknutím sa upravujú.

Výhody:

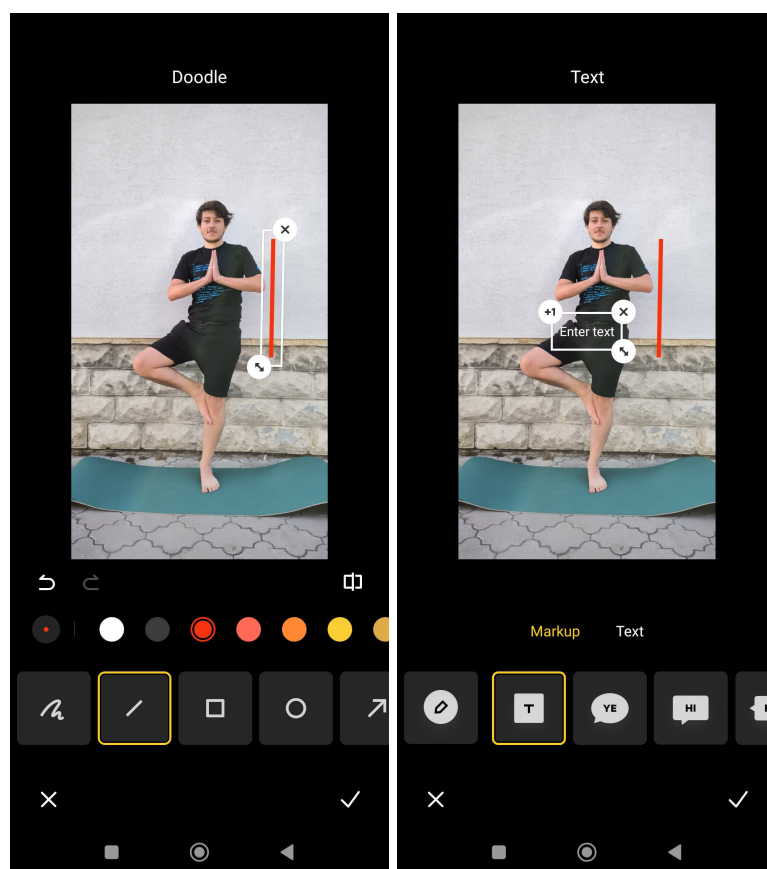
- Intuitívne gestá pre približovanie obrazu a zmenu veľkosti tvarov.
- Možnosť úpravy vektorov pomocou transformátora.

¹[r/yoga](#) – komunita pre zdieľanie príspevkov spojených s jogou.

- Minimalistické nastavovanie hrúbky čiar.

Nevýhody:

- Obsahuje veľa funkcií, ktoré nie sú potrebné.
- Obsahuje málo tvarov na značenie.
- Komentáre zaberajú veľkú plochu.
- Tlačidlo na zmazanie na mieste, kde by mohla byť druhá kotva. Na doladenie treba posunúť celý objekt hore a potom spodnou kotvou dotiahnuť dole.



Obr. 4.1: Snímky aplikácie na upravovanie obrázkov na MIUI 12. **Vľavo:** Palety pre tvorbu tvarov, nastavenie farby a hrúbky. **Vpravo:** Rôzne spôsoby komentárov.

4.3 Canva

Canva je aplikácia, ktorá ponúka jednoduché, no zároveň pestré možnosti úpravy fotografií a dokumentov. Užívateľom umožňuje vkladanie tvarov, voľné kreslenie pomocou pera a dokonca aj integráciu vektorovej grafiky a ďalších médií. Prvky sa na plátno umiestňujú kliknutím z ponuky dostupných elementov. Po kliknutí sa nový element zobrazí v strede plátna a je možné ho ďalej presúvať a upravovať podľa potreby. Rovnaký spôsob funguje aj pri pridávaní komentárov. Canva poskytuje najviac možností pri transformovaní elementov

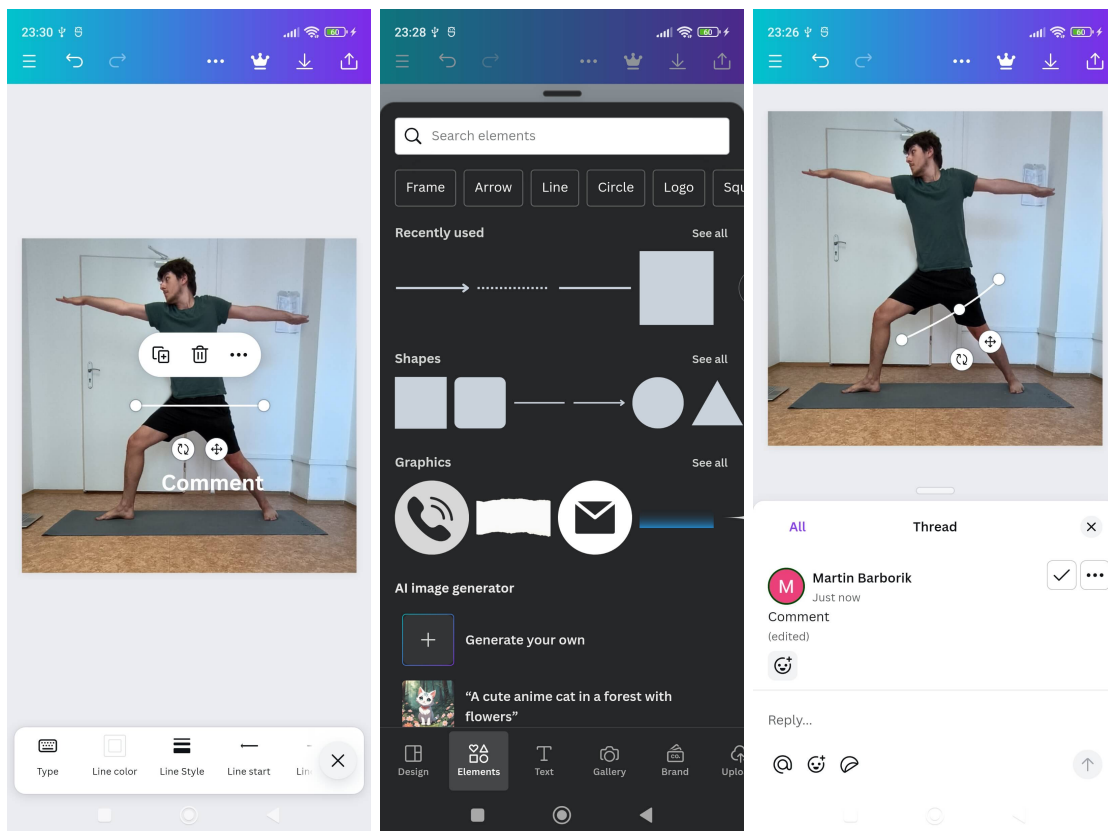
plátna. Transformátor pridáva k elementu kotvy, pomocou ktorých môže užívateľ tvar nahoväť aj prehýbať. Dodatočné kotvy a možnosti využívajú princíp proximity (vysvetlený v 3.3), okrem kotiev sa pozícia menu nikdy nemení pri rotovaní, sú preto vždy na predvídateľnom mieste. Počas voľného kreslenia aplikácia umožňuje zvoliť nástroj na výber prvkov, čo zabraňuje náhodnému kresleniu a umožňuje presné označovanie bez možnosti nežiaducich omylov. Ku každej existujúcej entite na plátne je možné pridať komentár. Táto funkcia je určená pre spoluprácu v aplikácii, avšak tento spôsob by bolo možné aplikovať aj pri komentovaní pozícií. Pokiaľ užívateľ použije gesto pinch a oddialenie dostatočne dlho, tak sa zobrazí galéria. Rozhranie pre editovanie je na obrázkoch 4.2.

Výhody:

- Komentáre, ktoré neblokujú obrázok.
- Rozhranie vhodné aj pre pokročilých užívateľov.
- Ukladanie práce na cloud, možnosť zdieľať.
- Spätné upravovanie objektov pomocou transformátora.

Nevýhody:

- Pre účely komentovania pozícií obsahuje priveľa nastavení, komponentov a médií.
- Umožňuje vloženie uhlov na plátno, ale nemožno nastaviť hodnotu, sú to statické obrázky.
- Vyžaduje úsilie napodobniť riešenia z kapitoly 2.1.



Obr. 4.2: Snímky grafického editora aplikácie Canva. **Vľavo:** Transformátor využíva proximitu. **Uprostred:** Paleta na výber elementov rôznych typov. **Vpravo:** Komentáre pod obrázkom.

4.4 Instagram

Populárna sociálna sieť na zdieľanie fotografií a videí, často používaná u inštruktorov. Poskytuje vytváranie príbehov, v ktorých je možné zvolenú fotku anotovať. V obrázku 4.3 vidieť nástroje, ktoré aplikácia poskytuje. Text sa najprv napíše, komentár je následovne po zadaní textu umiestnený do stredu obrazovky. Užívateľ ho môže premiestňovať. Kreslenie je formou voľnej ruky, jediný dostupný tvar je šípka. Aplikácia sa viac zameriava na rôzne štýly voľného kreslenia než na podporu viacerých tvarov. Užívateľ nemôže nakreslené tvary presúvať ani modifikovať, iba mazať. Mazanie je formou kroku späť alebo gummy, ktorá maže nakreslené tvary po úsekoch. Funkcionalita, ktorú poskytuje Instagram a ktorú som pridal aj do svojho rozhrania je schovanie tlačidiel počas kreslenia.

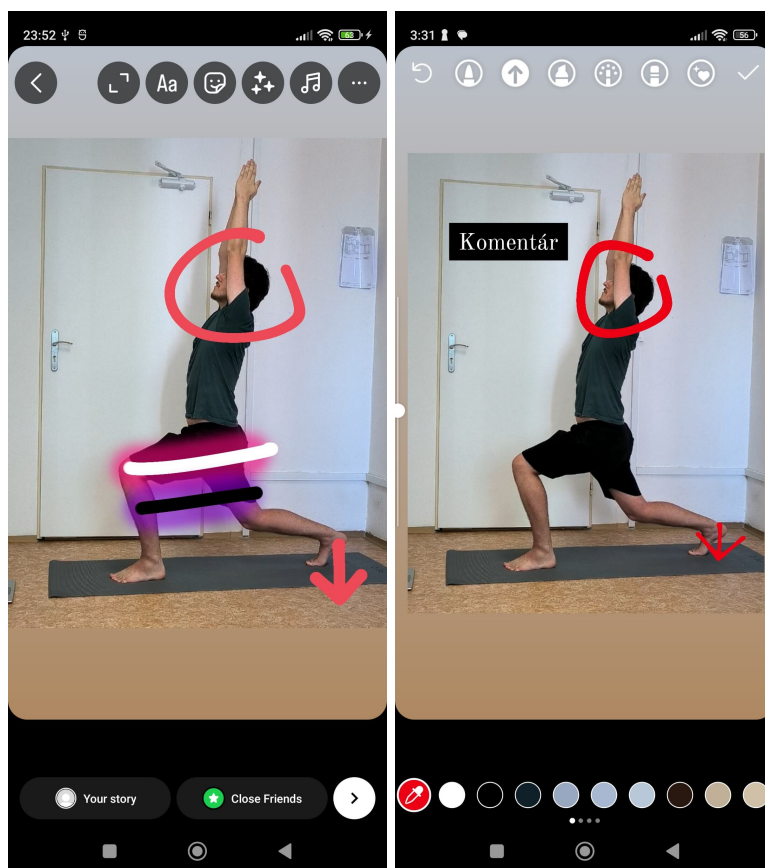
Výhody:

- Umožňuje rýchle naznačenie.
- Jednoduché zakreslenie ohybu.
- Možnosť použiť zvýraznené čiary.
- Čistejší vzhľad bez hornej lišty.

- Schovávajúce tlačidlá počas kreslenia.

Nevýhody:

- Nepodporuje modifikáciu existujúcich objektov na plátne, okrem textu.
- Tlačidlá na hornej lište nie sú veľmi ľahko prístupné na väčších zariadeniach.
- Nepodporuje redo, iba undo, tým pádom pri situácii, kde sa zmaže omylom viac objektov, je potrebné ich nakresliť znovu.
- Zlá viditeľnosť horných tlačidiel v prípade bieleho alebo svetlého obrázka.
- Limitovaný na jednu platformu.
- Podporuje zväčšovanie textu pomocou odtiahnutia prstov od seba, ale pri miernom prekryvaní dvoch objektov nejde modifikovať spodný.



Obr. 4.3: Snímky upravovania fotiek pri vytváraní príbehov v aplikácii Instagram. Rozloženie nemá hornú lištu, namiesto toho sú tlačidlá priamo na fotke. **Vľavo:** Rôzne druhy zvýraznenia nakreslených tvarov. **Vpravo:** Voľby farieb, nástrojov a vložený komentár.

4.5 Picmarker

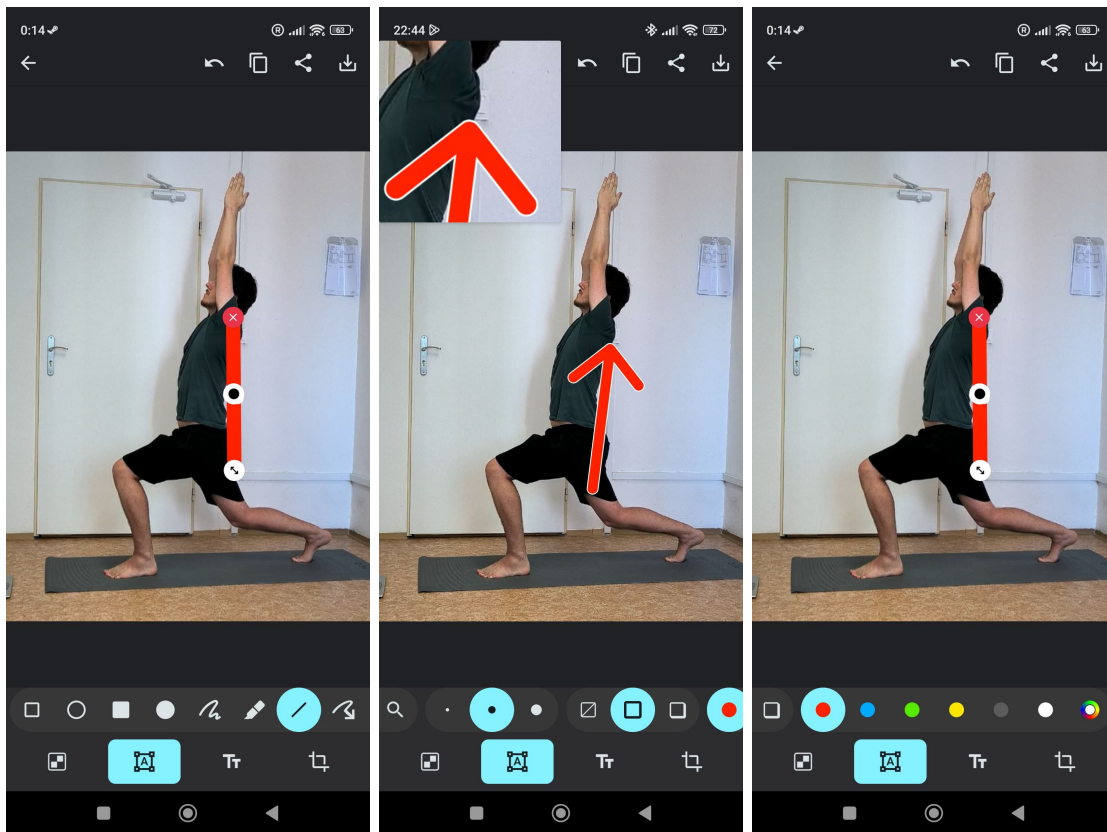
Picmarker je mobilná aplikácia na anotáciu obrázkov. Na rozdiel od ostatných aplikácií má všetky tvary, hrúbky a farby v jednom rade, čím sa uvoľňuje priestor nad paletou. Poskytuje kreslenie aj voľnou rukou, aj tvarmi. K tvarom pridáva kotvu na ohýbanie, natiahnutie tvaru a mazanie. Textový komentár sa vkladá zvolením nástroja v spodnej lište a následným kliknutím na miesto, kde komentár chceme vložiť. Použitie aplikácie je vidieť na obrázkoch 4.4. Picmarker som ukazoval aj účastníkom testov, zaujímalo ma hlavne ako reagovali na ohýbanie krivky kotvou uprostred a na rozloženie tlačidiel.

Výhody:

- Aplikácia zobrazuje v malom okne miesto, kde sa aktuálne kreslí, vidieť presne kam ide vrchol.
- Upravovanie existujúcich objektov.
- Okraje okolo tvarov zabraňujú splývaniu s pozadím.
- Zobrazenie okna pri posúvaní kotvy pomáha s presnosťou.

Nevýhody:

- Nastavenie hrúbky čiar a farieb je ľahko prehliadnúť.
- Podporuje len krok späť.
- Na zmenu farieb treba vždy pred zvolením alebo po nakreslení posúvať spodnú lištu do strany.
- Tlačidlo na zmazanie na mieste, kde by mohla byť druhá kotva. Na doladenie treba posunúť celý objekt hore a potom spodnou kotvou dotiahnuť dole.



Obr. 4.4: Snímky aplikácie Picmarker. **Vľavo:** Paleta všetkých nástrojov. **Uprostred:** Voľba hrúbky a ohraničenia čiar, počas presúvania kotvy na konci šípky sa zobrazí malé okno. **Vpravo:** Voľba farieb, zobrazenie šiestich výrazných farieb. Nástroje a všetky možnosti v jednom rade, pri prepínaní tvarov a farieb je nutné posúvať paletu, počas testu sa stalo, že účastník neobjavil časť palety s farbami.

4.6 CoachNow

CoachNow je mobilná aplikácia navrhnutá pre trénerov a športovcov určená pre komunikáciu, poskytnutie rád a plánovanie tréningov. Tréner vytvorí skupinu, do ktorej môže on alebo športovci posilať príspevky. Podporuje komentovanie príspevkov pomocou editora z obrázku 4.5. Na rozdiel od predošlých riešení ide o komentovanie pomocou nahrávania videa, počas ktorého sa môžu do fotky kresliť rôzne tvary. Tvary na plátne je možné transformovať, ale nie je možné meniť farby ani hrúbku existujúcich tvarov. Farby sú nastavené pre každý tvar individuálne. Mazanie je možné cez krok späť alebo vymazanie celého plátna. Uživatelské rozhranie pre komentovanie je na obrázkoch 4.5.

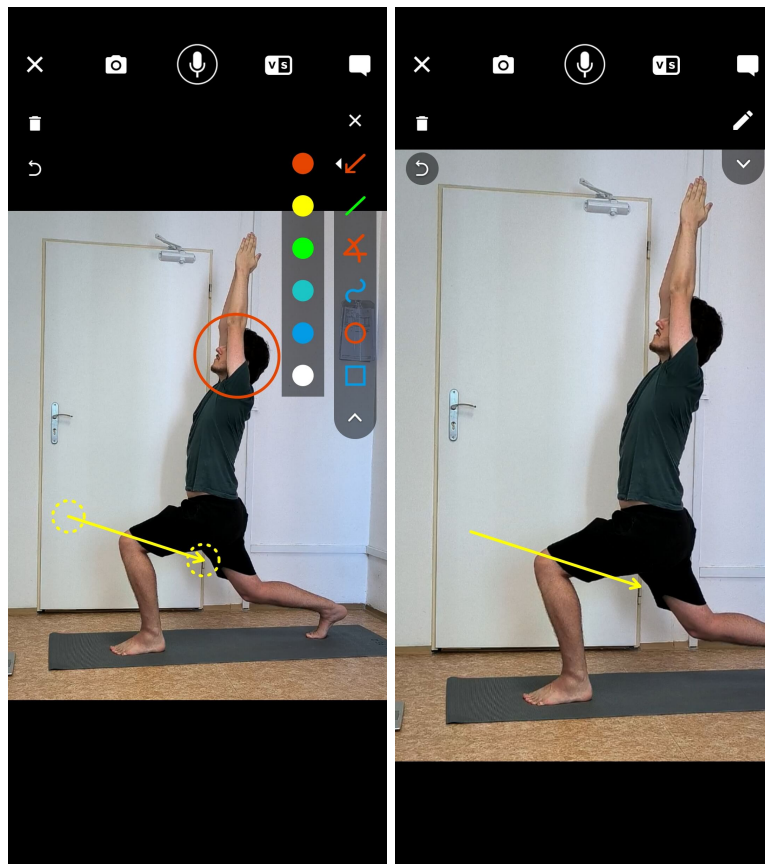
Výhody:

- Umožňuje komunikáciu medzi trénermi a športovcami dodatočnými komentármi na svojej vlastnej sieti zameranej na tréningy.
- Nástroje na ľahko prístupnom mieste.
- Spätné upravovanie objektov pomocou transformátora.

- Nástroj na kreslenie uhlov.

Nevýhody:

- Obmedzené možnosti úprav tvarov, vrátane farby a hrúbky.
- Tlačidlo na mazanie vyčistí celé plátno.
- Potreba nahrávania videa môže byť menej praktická pre rýchle komentáre a u niektorých osôb nepohodlná.



Obr. 4.5: Snímky komentovania formy v aplikácii Coachnow. **Vľavo:** Farba sa nastavuje individuálne pre každý nástroj, transformátor nemá vyplnenie. **Vpravo:** Schovanie paliet.

Kapitola 5

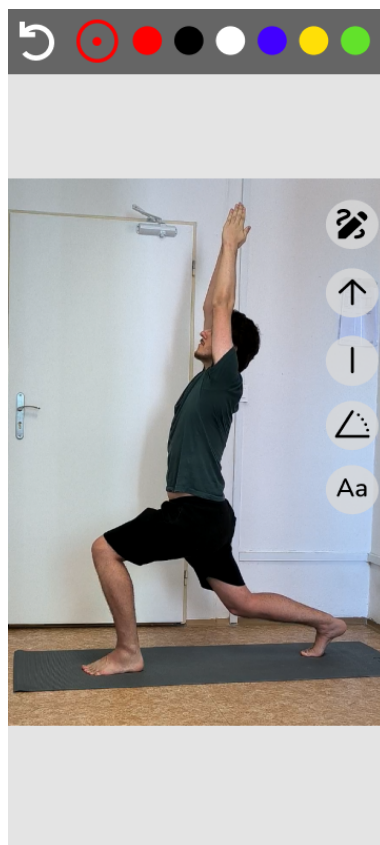
Návrh užívateľského rozhrania: prototypy a iteratívne vylepšovanie

Pri vývoji som priebežne od začiatku testoval dielčie časti rozhrania s účastníkmi, pri ktorom som používal odporúčania a postupy popísané v predchádzajúcej kapitole (vid. kapitola 3.1). Keďže niektoré prvky rozhrania neboli rozsiahle, tak som vykonával testy na viacero prvkov za sebou. Prvotných testovaní sa zúčastnil mix osôb cieľovej skupiny a mimo cieľovej skupiny, zámerom tohto prístupu bolo šetriť si ochotných odborníkov na testovanie uceleného rozhrania vychádzajúceho z otestovaných prototypov. Na základe výsledkov týchto testov som pravidelne upravoval a vylepšoval svoje návrhy. V tejto kapitole podrobne rozoberám jednotlivé prvky rozhrania a opisujem, ako sa na základe iteratívneho testovania vyvíjali. Prototypy som tvoril v aplikácii Figma a viac interaktívne prvky som implementoval za účelom testovania v prototypoch implementovaných v Reacte. Zvolil som formu iteratívne testovať menšie prototypy ako navrhnúť jeden celok a ten otestovať. Experimentoval som. Príklad prototypu z testu je na obrázku 5.1.

5.1 Spôsob kreslenia

Počas testovania rôznych aplikácií vo fáze prieskumu existujúcich riešení som našiel niekoľko opakujúcich sa techník na vytváranie tvarov na plátne. Tieto štyri techniky sa ukázali byť najčastejšie:

- **Ťahanie a pustenie (Drag and Drop)** – Umožňuje používateľom vytvoriť objekt potiahnutím objektu z palety na miesto, kam ho chcú vložiť. Toto riešenie mala napríklad desktopová verzia aplikácie Canva.
- **Kliknutie na umiestnenie (Click to place)** – Používatelia kliknú na požadované miesto na plátne, kde sa zvolený objekt umiestni. Tento spôsob sa prevažne používa pri zadávaní textových komentárov. Príkladom použitia je PicMarker.
- **Kliknutie na vytvorenie (Click to create)** – Kliknutím na nástroj v palete sa uprostred plátna vytvorí zvolený objekt. Mobilná verzia aplikácie Canva používa tento spôsob na vkladanie všetkých objektov, zatiaľ čo iné ho používajú skôr pri vkladaní textových komentárov.
- **Kliknutie a ťahanie (Click and Drag)** – Kliknutím a ťahaním sa na plátne vytvorí objekt podľa trasy ťahania.



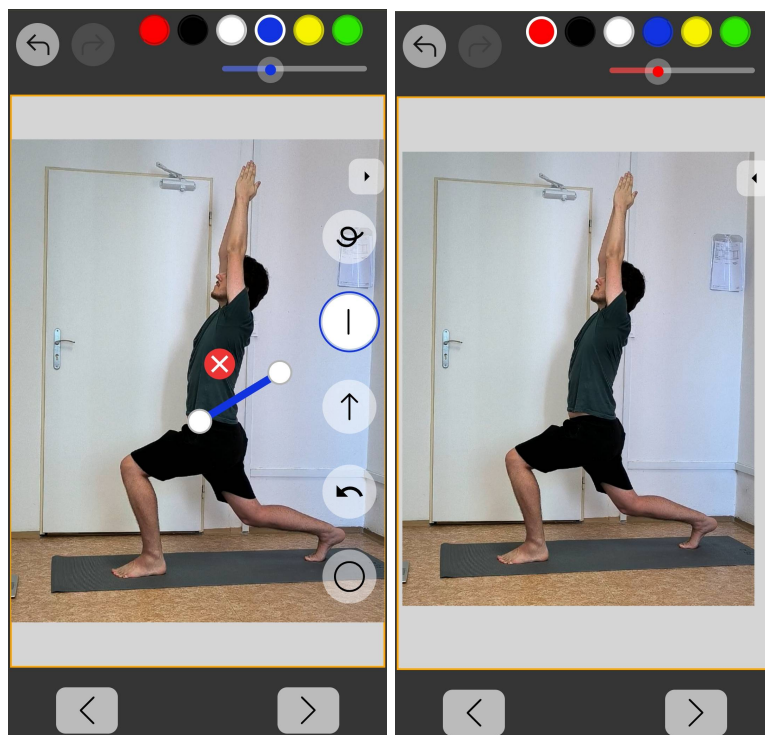
Obr. 5.1: Prototyp vo Figue na testovanie rozloženia, palety farieb, kroku späť a panela nástrojov s ďalšími tvarmi prístupnými v modálnom okne vedľa tlačidla po dvojkliku.

- **Voľné kreslenie (Freehand)** – Kreslenie priamo na plátno voľnými ťahmi, podobné kresleniu reálnou ceruzkou.

Z popísaných spôsobov som do prototypov v Reacte zvolil kliknutie na mieste v plátno na vytvorenie textového komentára, kliknutie a ťahanie na kreslenie všetkých tvarov a voľnú kresbu na ohyby pomocou špičiek alebo kreslenie viac špecifických tvarov. Počas testov bolo možné pozorovať satisficing (v sekcii 3.2) aj pri kreslení. Pri poskytnutí kresby s voľnou rukou si užívateľ označil všetko týmto spôsobom. Zatiaľ, čo iný užívateľ chcel doladovať vzhľad a preto využil viacero tvarov tvorených z iných nástrojov.

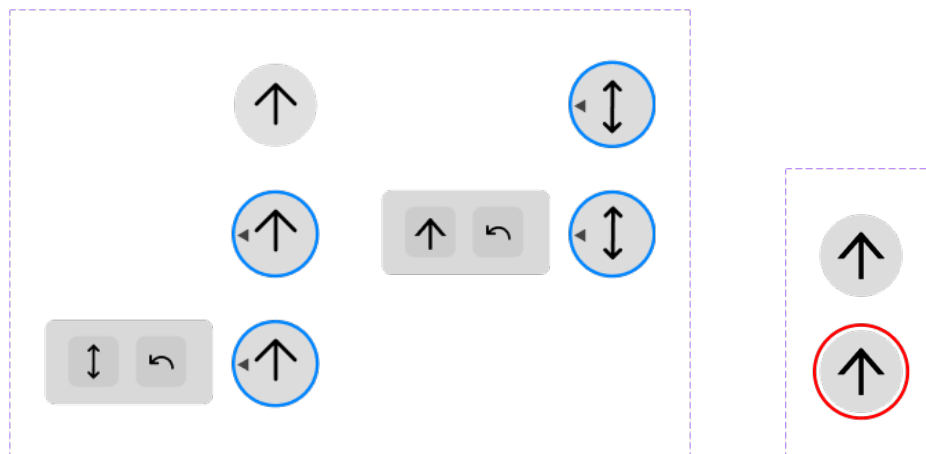
5.2 Panel s nástrojmi

Návrh panela s nástrojmi bol náročný. V prvých fázach som testoval aplikácie spomenuté v kapitole 4. Podľa toho som postupoval pri rozložení a výbere nástrojov. Možností tvarov, z papierovej formy, bolo veľa. Aj po zredukovaní možností bolo kombinácií ako urobiť panel veľa. Chcel som vyskúšať využitie gest a schovať viac špecifické nástroje do okna s variantami. Napríklad šípka mohla byť obyčajná, obojstranná alebo kreslená voľnou rukou. Prvý snímok z obrázkov 5.3 zobrazuje prechody medzi stavmi tlačidla. Spolieha sa na využitie dvojkliku. Pri prvom kliku sa označí nástroj za aktívny a pri druhom sa otvoria možnosti pre iný variant podobného tvaru. Testy nedopadli optimálne. Extra možnosti pre nástroje



Obr. 5.2: Prvý prototyp v Reacte na testovanie kreslenia. Podľa Steva Kruga nie je skoro na testovanie. V tomto prototypy som testoval základné tvary, možnosť schovania panela s nástrojmi a základné transformovanie čiar a kreslenie kruhu. Hlavné poznatky z testu boli, schovávanie panela s nástrojmi cez tlačidlo neskúšali. V prípade, že sa miesto označenia prekryvalo s tlačidlom, tak si obrázok priblížili. Tlačidlo vrátenia kroku späť pôsobilo ako tlačidlo na prechod na predošlú stranu.

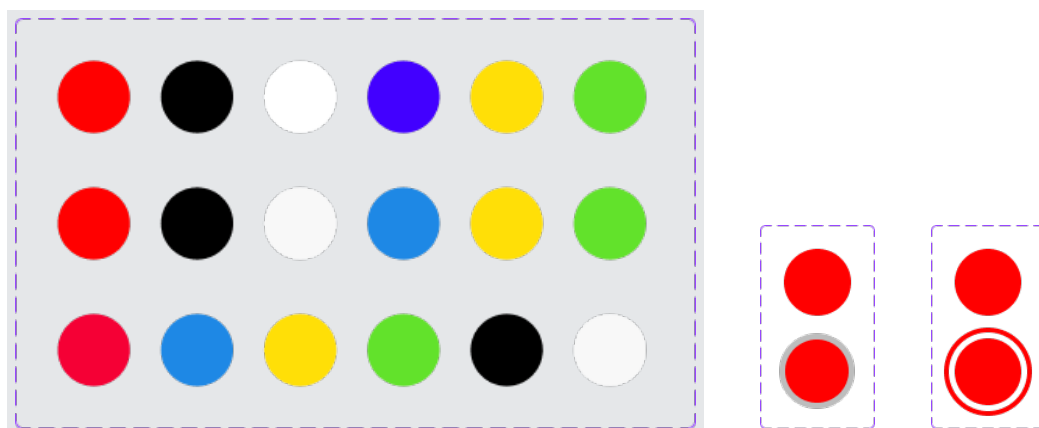
neboli nájdené. Návrh bol upravený na jeden vertikálny panel s väčším počtom tlačidiel, ale limitovanejším počtom možností na kreslenie tvarov. Pôvodne bola farba na výber tlačidla statická, ale jedna osoba z testovania naznačila, že by očakávala zmenu farby označenia tlačidla podľa aktuálne zvolenej farby v palete farieb, napríklad červenej ako na druhom snímku obrázku 5.3. Túto pripomienku som zakomponoval vo všetkých budúcich verziách. Počas interakcie s objektami plátna sa panel schováva. Po testoch som sa účastníkov pýtal na názor na schovávanie panelu. Väčšina si ani neuvedomila, že zmizol. Prírodzene kreslili a po ukončení kreslenia pokračovali ďalej.



Obr. 5.3: Komponenty Figmy na zvolenie nástroja na kreslenie. **Vľavo**: Verzia s limitovaním počtu tlačidiel na obraze pomocou kategorizovania, so statickou farbou označenia. **Vpravo**: Zobrazenie nástroja priamo bez ďalších okien. Farba označenia je totožná s aktuálne zvolenou farbou v palete farieb.

5.3 Výber farieb

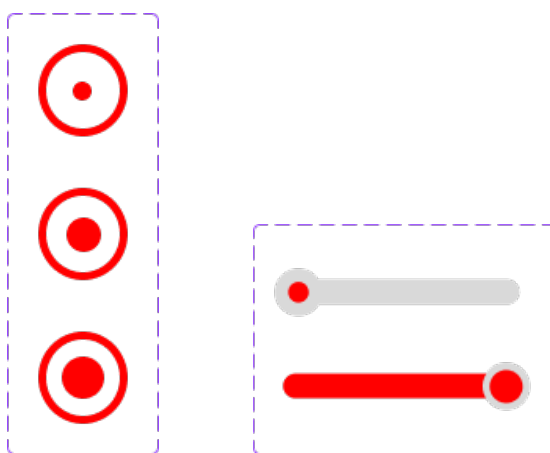
Paleta s farbami v aplikáciach na značenie fotiek ako napríklad Picmarker alebo Coachnow poskytujú menšie množstvo možností. Rovnako aj pri komentovaní na papier mali účastníci päť farebných fixiek a modré pero, ale väčšina sa držala jednej až štyroch farieb. Z tohto dôvodu som do palety v rozhraní tejto aplikácie zvolil tiež šesť farieb a to rovnakých šesť farieb ako v papierovej úlohe. Nezvolil som iba štyri farby. Aj keď sa nepoužijú všetky, použijú sa rôzne kombinácie, tak ako na papieroch. Komponenty pre výber farieb sú zobrazené na obrázkoch 5.4.



Obr. 5.4: Komponent palety farieb obsahuje šesť tlačidiel. **Vľavo**: Tri varianty pod sebou, postupné zmeny v palete farieb od počiatocnej hore po aktuálnu dole. **Uprostred**: Predošlé označenie aktuálnej farby. **Vpravo**: Aktuálne označenie vybranej farby. Inšpirované z aplikácie 4.2.

5.4 Nastavenie hrúbky tvarov

Na obrázku 2.1 z kapitoly o výskume na papieroch je vidieť šípku na naznačenie posunu nohy vpred. Je to jeden z príkladov, v ktorých je cvičiacemu zdôraznené väčším tvarom, čo má robiť. Na zakomponovanie tohto efektu bez pridania špeciálnej šípky podobnej tej z obrázku, je možné pridať komponent, ktorý bude ovládať hrúbku šípky a aj ostatných tvarov. Prvá verzia, s ktorou som vykonával test bola inšpirovaná riešením v editore od MIUI 4.2. Je to jednoduché riešenie, ktoré nezaberá veľa miesta na obrazovke. Užívateľ klikaním prepína medzi niekoľkými hrúbkami. Po teste som od tohto riešenie prešiel na iné. Účastníci testovania sa s týmto spôsobom nestretli. Väčšina si myslela, že ide o spustenie nahrávania obrazu, než o nastavovanie hrúbky. Pri otázke, na aký spôsob sú zvyknutí odpovedali slider. Účastníci ďalších testov s týmto riešením nemali problém a vedeli hneď k čomu slúži. Obe riešenia sú zobrazené na obrázku 5.5.

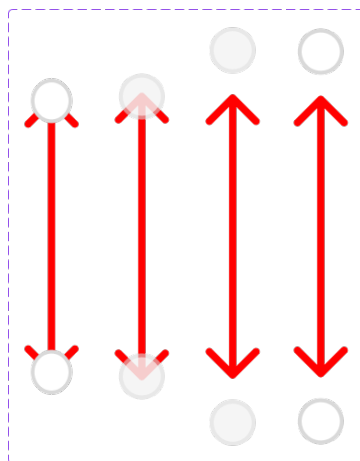


Obr. 5.5: Komponenty Figma na nastavovanie hrúbky tvarov na plátne pomocou klikania vľavo, neskôr nahradeného sliderom vpravo z dôvodu podobnosti k ikonke nahrávania prvotného návrhu.

5.5 Transformátor

Transformátor je komponent na modifikáciu existujúceho objektu na plátne pomocou posúvania kotiev. Keďže Instagram bol medzi užívateľmi veľmi populárny a nakreslené tvary sa v ňom nedajú presúvať ani transformovať, najprv som uvažoval o potrebe transformátora. Počas testov som si však všimol, že na doladenie tvaru užívateľa objekt zmazali a snažili sa znovu nakresliť objekt lepšie, alebo ho skúšali hneď presunúť. Pozorovaním som sa uistil o potrebe transformátora. Aspoň u preddefinovaných tvarov ako napríklad kruh.

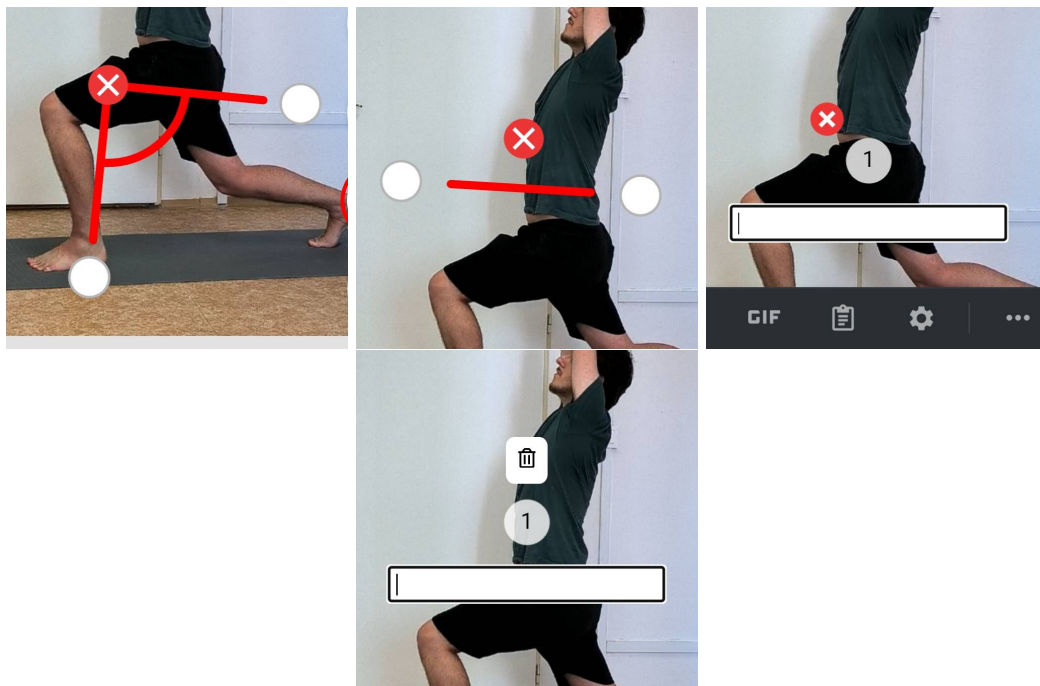
Prvotný prístup bol transformátor zobraziť len po označení objektu, ktorý chce užívateľ modifikovať. Pri tomto prístupe došlo k situáciám, kde účastník testu opakovane prekresloval tvar, ktorý sa mu nepodarilo nakresliť tak ako chcel, lebo mu nenapadlo na objekt kliknúť a použiť kotvu na doladenie. Pre zabezpečenie, že užívateľ objaví túto funkciu, je novovytvorený objekt ihneď označený. Tým sa na ňom zobrazí aj transformátor. Po zmene, nasledujúce testy tento problém už nemali. Forma transformátora sa vyvíjala, postup je priblížený na obrázku 5.6.



Obr. 5.6: Varianty transformátora nakreslených tvarov. Od prvotného vľavo po finálny vpravo. Prvý spôsob vyžadoval zrušenie označenia objektu, aby bol špic viditeľný. Prešiel som teda na priehľadnú kotvu, ktorá vyriešila blokovanie, avšak nie počas transformovania tvaru. Tento problém riešila aplikácia PicMarker (sekcia 4.5), ktorá poskytla pohľad cez malé okno. Namiesto nového okna som skúsil jednoduchší spôsob a to posunúť kotvy od objektu a spoliehať sa na proximitu (sekcia 3.3). Po zmene od priehľadnej kotvy na plnú farbu, pretože si kotvy niektorí nevšimli, boli názory na toto riešenie pozitívne.

5.6 Mazanie objektov

Na mazanie som v aplikáciách videl tri spôsoby. Krok späť, guma a tlačidlo s krížikom alebo košom. Krok späť je nevyhnutná funkcia pri kreslení, najmä pre vrátenie omylom zmazaného objektu. Gumovanie prináša inkonzistenciu. Sú dva prístupy ako gumovanie funguje. Pri prejdení objektom sa zmaže malý úsek. Druhý je zmazanie celého objektu. Najkonzistentnejšie je tlačidlo, ako to robia aplikácie Canva (sekcia 4.5) a Picmarker (sekcia 4.5). V testoch s Picmarker malo tlačidlo na mazanie na začiatku kresleného objektu zlý vplyv. Užívateľ pri úprave presúval objekty menej prirodzeným spôsobom. Vyhýbal sa veľkému krížiku. Skúšal som preto vo vlastných prototypoch iné spôsoby. Umiestnenie krížika do stredu objektu. Tlačidlo v prípade červených objektov zapadlo s objektom. Zároveň sa užívateľ teraz vyhýbal stredu objektu, aby nestlačil krížik. Druhý pokus bol oddialiť ho a spoliehať sa na proximitu. Lepší výsledok. Užívatelia plne využívali kotvy aj presúvali priamo objekt bez kotiev. V tomto prístupe sa hýbalo tlačidlo spolu s objektom. Pokiaľ sa objekt natočil, tak sa hýbalo tlačidlo s ním. Účastníci testu ho museli niekedy hľadať. Finálny spôsob je udržiavanie tlačidla od objektu o určitú vzdialenosť buď nad ním alebo pod, v prípade, že je objekt moc vysoko na zobrazenie dodatočného tlačidla nad ním. Taktiež som zvolil viac neutrálnu farbu a ikonku. Červený krížik je výrazný, užívatelia si ho všimnú hneď, ale zároveň sa naň sústredia počas kreslenia, neutrálné, minimalistické riešenie ako v aplikácii Canva neodradilo užívateľov, našli ho hneď a neupútavalo pozornosť, ktorú by mal inštruktor venovať cvičencovi v pozadí. Ukážky jednotlivých spôsobov na obrázkoch 5.7.

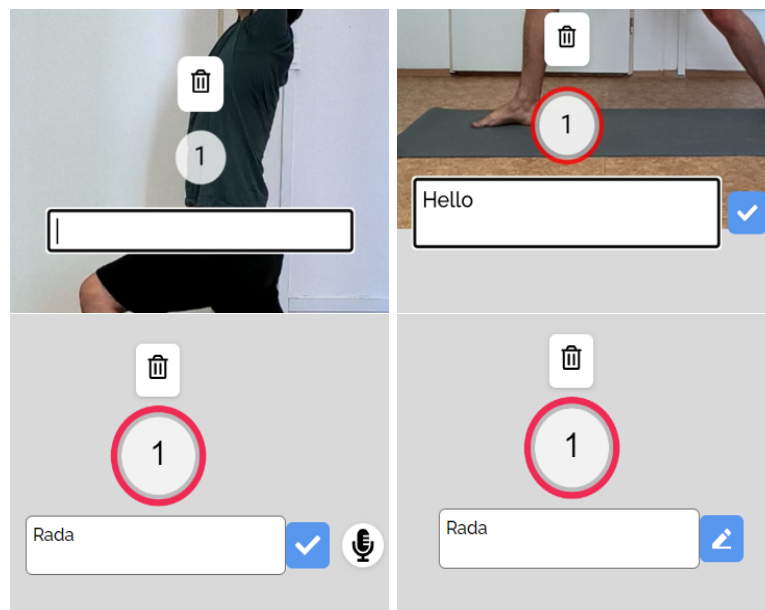


Obr. 5.7: **Horný rad:** Mazanie červeným tlačidlom. Skúšaných bolo niekoľko spôsobov. Líšila sa vzdialenosťou od objektu, všetky boli však závislé od otočenia objektu. **Spodný:** Riešenie na predvídateľnejšom mieste, neotáča sa s objektom.

5.7 Textový komentár

Zakomponovanie samotného textu ako to robia aplikácie, ako napríklad Instagram, priamo do fotky so zachovaním čitateľnosti na mobilnom zariadení spôsobuje prekrytie príliš veľkej plochy. Preto som hľadal rôzne spôsoby ako tento problém vyriešiť. Je možné text oddeliť od plátna úplne a zobrazíť ho pod ním. Rozhodol som sa však vyskúšať interaktívne hotspoty. Ide o malé výrazné a niekedy aj pulzujúce tlačidlá vložené priamo do fotky, ktoré po kliknutí zobrazia okno s ďalším obsahom. Prvý návrh tvorili jednofarebné klikateľné hotspoty. Tento prístup fungoval, proximita k označeným častiam tela stačila. Problém nastal len v prípadoch, keď boli dva hotspoty príliš blízko. Riešením bolo pridať farebný okraj k hotspotu totožný s farbou nakresleného tvaru ku ktorému spadal, podobne ako okomentovaný papier na obrázku 2.2 z kapitoly s výskumom. Otvorením okna na písanie textu, hneď pri vytvorení hotspotu, bolo užívateľom jasné, že sa po uzavretí dá kliknutím na hotspot znovu otvoriť.

Prvotný návrh podporoval len písanie textu pomocou klávesnice zariadenia. Počas testovania tohto prístupu mi účastník povedal, že v aplikáciách primárne používa funkcionality prevádzania reči na text. Tento spôsob umožňuje urýchliť komentovanie a zároveň zachovať formu textového komentovania pre cvičencov, ktorí preferujú neprehrávať si zvukové správy. Jediná nevýhoda je potreba vetu zopakovať alebo upraviť v políčku pre vstup v prípade nesprávneho spracovania prirodzeného jazyka.



Obr. 5.8: Komponent na pridávanie komentárov na fotku bez veľkého prekryvania. V pokoji ide o kruh, ktorý po rozkliknutí poskytne užívateľovi vstupné pole. **Vľavo hore:** Prvá verzia komentára inšpirovaná z obrázka 2.3. **Vpravo hore:** Zakomponovanie farby a pridanie tlačidla na potvrdenie dokončenia písania. Účastníci testu, ktorí nestlačili klávesu enter nevedeli ako dokončiť komentár, to bolo dôvodom pre túto úpravu. **Vľavo dole:** Pridanie generovania textu pomocou prekladu reči do textu. **Vpravo dole:** Po dokončení textu sa prepne hotspot do módu na zablokovanie omylného nahrávania zvuku, ktoré by mohlo zmazať text. Zároveň tlačidlo nastaví kurzor na písanie na koniec textu. Užívateľ môže hneď pokračovať, ak by chcel.

5.8 Zvýraznenie nakreslených tvarov

Počas testovaní došlo k prípadom, keď tvar bez okraja splýval s fotkou v pozadí. Preto som skúšal rôzne spôsoby ako tvary zvýrazniť. Experimentoval som preto s tieňmi a okrajmi na viacerých fotografiách, ktoré obsahovali odtiene poskytnutých farieb v palete. Vyskúšané spôsoby sú viditeľné na obrázku 5.9 a finálne ohraničenie na obrázku 5.10.

- **Tiene** – Tieni rovnakých alebo podobných odtieňov zvolenej farby vytvorili efekt, pri ktorom vyzerali čiary hladšie, ako keby boli nakreslené ceruzkou, avšak problém nevyriešili. Ani tieň opačnej farby neboli dostatočne výrazné.
- **Biele vyplnenie s tieňom zvolenej farby** – Spôsob aký poskytuje aplikácia Instagram (vid. sekcia 4.4). Objekt je výraznejší ako predošlé riešenie, ale farba pomocou ktorej sa odlišujú objekty sa stráca ako v pôvodnom prípade.
- **Okraj tmavšieho odtieňa zvolenej farby** – Pridanie okraja plnej o niekoľko odtieňov tmavšej farby nemali výrazný efekt na vyriešenie problému.
- **Okraj statickej neutrálnej farby** – Aplikácia Picmarker používa dve neutrálne farby, bielu a čiernu, výber farby okraja je založený od farby objektu. Vo svojej aplikácii pri malom počte farieb som chcel nájsť viac konzistentné riešenie. Zvolil som farbu medzi bielou a čiernou, šedú.



Obr. 5.9: Experimentovanie s knižnicou Konva so zvýrazňovaním tvarov na fotke pre zabránenie splývania s pozadím. **Značka 1:** Tieň. **Značka 2:** Okraj tmavšieho odtieňa zvolenej farby. **Značka 3:** Bez modifikácie. **Značka 4:** Biele vyplnenie s tieňom zvolenej farby. **Značka 5:** Okraj statickej neutrálnej farby.

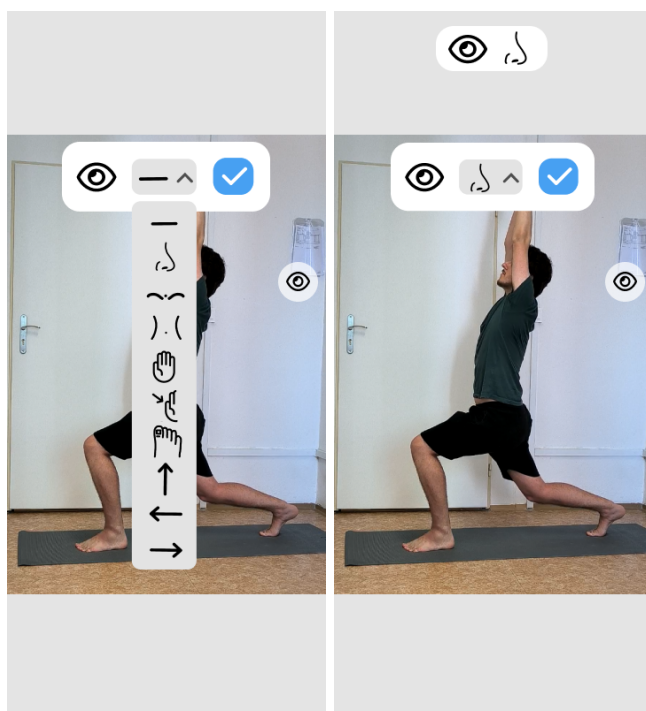
5.9 Zameranie pohľadu

V kapitole 2 je priblížené zameranie pohľadu v joge. Z popisu vidieť, že je počet bodov pohľadu limitovaný, existuje deväť hlavných bodov zamerania. S touto informáciou je možné navrhnúť niekoľko spôsobov ako uľahčiť naznačenie pohľadu na fotografii bez potreby textového vstupu. Počas výskumu jogy som, okrem textových popisov, našiel niekoľko ilustrácií používajúcich ikonky. Vytvoril som na otestovanie použiteľnosti prototyp vo Figma, niekoľko snímok z neho je na obrázkoch 5.11. Výsledky neboli pozitívne. Riešenie malo niekoľko problémov. Ikonky nemusia byť hneď samozrejme a neboli ani v danom kontexte pre inštruktora. Fotografia s označením som ukázal aj účastníkom, ktorí nespádajú pod inštruktora a mohli by byť na prijímacej strane. Ani tam sa ikonkám nedarilo. S ďalším nastudovaním som odhalil nedostatok a to, že sa okrem deviatich pohľadov robia modifikácie, napríklad pri polohe pes by sa mal cvičenec pozeráť na svoje brucho.

Jeden inštruktor mi naznačil na papier z výskumu v kapitole 2.1, pohľad na koleno, zistil som, že sa robia modifikácie aj pohľadov, nie len pozícií. V novších papieroch bolo viacero značení pohľadu. Jednoducho okom a šípku. Do fotografií som vo Figma vkladal inštancie komponentu oka so šípku. Zo šípok bolo účastníkom zrejmejšie, že ide o smer pohľadu. Tento prístup zostal aj vo finálnom riešení na obrázku 6.3.



Obr. 5.10: Finálna zvolená metóda a farba pre ohraničenie tvarov s účelom zabránenia splývania objektu s pozadím. Neutrálna farba medzi bielou a čiernou bola dostatočne kontrastná so všetkými šiestimi farbami palety a umožnila viditeľnosť na rôznych testovaných pozadiach.

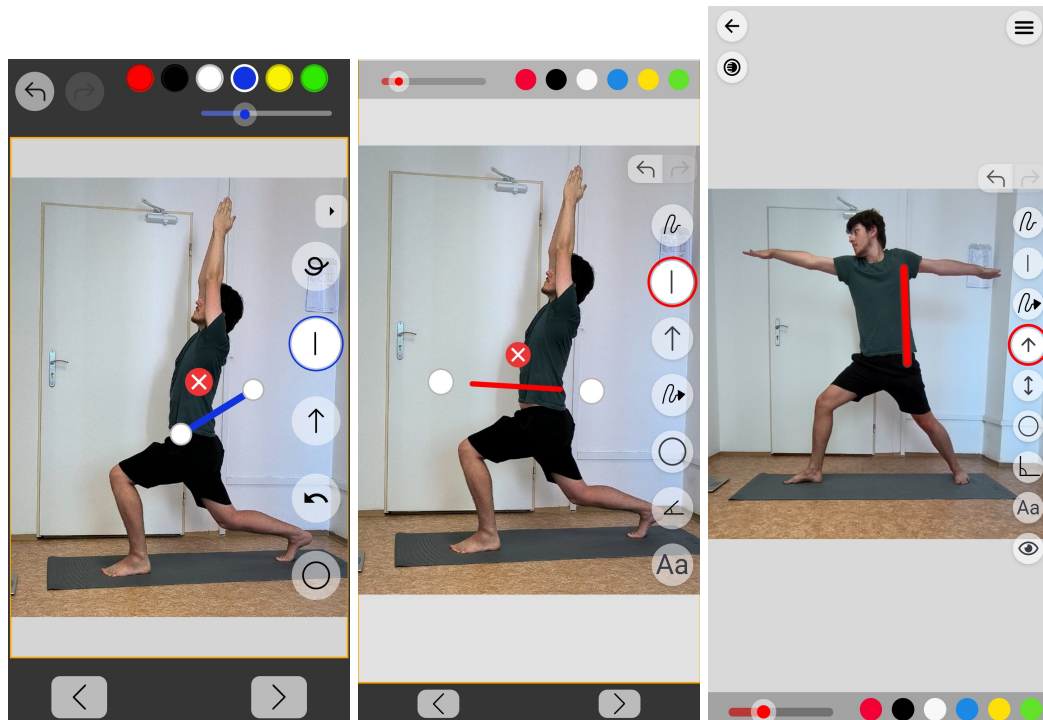


Obr. 5.11: Experimentovanie s prototypom na zamerania pohľadu vo Figma. Užívateľ si môže vybrať jeden z deviatich možných pozícií na zameranie pohľadu. Po zvolení sa na plátno pridá indikátor zvoleného pohľadu. Nedostatok tohto riešenia sa objavil počas testovania. Jednému z inštruktorov, ktorému som riešenie ukázal prišiel na účel, druhý nie. Po ukázaní iba príkladov fotografie s označením, nevedel účastník čo má podľa obrazu robiť.

5.10 Rozloženie

Problémom návrhu bolo nájsť vhodné rozloženie, ktoré by bolo estetické, ale aj efektívne. Počas testovaní som pozoroval, aké veľké tlačidlá môžu byť než začnú ovplyvňovať plátno za nimi. Prvé rozloženie nebolo esteticky doladené. Išlo o demonštračné rozhranie s účelom skúšať iné prvky. Počas testov som pozoroval ako sa užívateľom klikalo. Kedy vyžadovali

natiahnúť moc prst, využiť oba prsty. Chcel som pomocou panelu na pravej strane obmedziť potrebu dotyku rohu presne oproti miestu interakcie dominantnej ruky. Medzi verziami som pridával tlačidlá podľa tvarov, ktoré som videl z papierového výskumu sekcie 2.1. Hlavné ovplyvnené miesta boli vrchy zariadení. V dnešnej dobe sú niektoré mobilné zariadenia veľké. Všimol som si že horné časť obrazovky by mala obsahovať menej frekventované tlačidlá. Hlavne po zistení ako inštruktori zapájali farbu viac aj pri digitálnej verzii.



Obr. 5.12: Rozloženia v rôznych časoch. **Vľavo:** Tlačidlá undo a redo pôsobili ako tlačidlo na vrátenie na predchádzajúcu stránku. **Uprostred:** Premiestnenie tlačidiel histórie nad tlačidlá nástrojov a zmenšenie ich veľkosti. **Vpravo:** Premiestnenie palety na úpravu vzhľadu objektov na spodok s farbami bližšie k rohu. Zmena veľkosti bola využívaná rádovo menej ako farby. Pridanie tlačidiel na otvorenie menu, vrátenie sa do galérie fotiek a testovanie pridanie tlačidla na zvýšenie priehľadnosti objektov.

Kapitola 6

Implementácia a zhodnotenie

Táto kapitola popisuje implementáciu častí navrhnutého užívateľského rozhrania vo forme webovej aplikácie 6.1. Implementácia je z dôvodov vysokej interaktivity pomerne rozsiahla, preto sa v tejto kapitole zameriam na dôležitejšie a zaujímavejšie prvky rozhrania. Vychádzať bude z technológií popísaných v sekcii 3 aplikovaných v kóde.

6.1 Implementácia

Rozhranie je implementované ako progresívna webová aplikácia napísaná v JavaScriptovej knižnici React, zostavená pomocou nástroja Vite a jeho rozšírenia Vite PWA Plugin. Pridaním manifestu so základnými údajmi aplikácie ako názov, popis, ikony, je možné aplikáciu nainštalovať. Na všetku interakciu s plátnom som použil knižnicu React Konva.

6.1.1 Štruktúra aplikácie

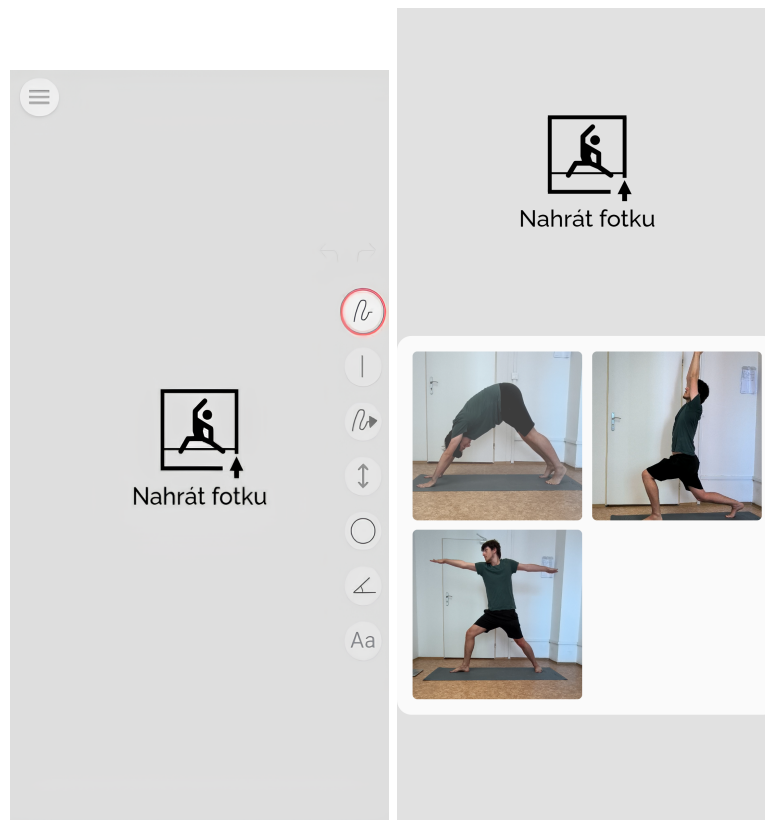
Aplikácia sa delí na dva pohľady `<Gallery/>` a `<Tool/>`. Prechod medzi nimi je zabezpečený pomocou knižnice React Router. Prvá časť, `<Gallery/>` je vstupnou bránou do aplikácie, skladá sa z komponentu na nahrávanie nových fotiek do aplikácie a galérie uložených fotografií vo forme mriežky. `<Tool/>` obsahuje hlavné funkcie aplikácie. Načítava a pripravuje dáta, je v ňom rozloženie plátna `<Canvas/>` a paliet `<Toolbar/>`, `<StylePalette/>`.

6.1.2 Spravovanie stavov

Na lokálne udržiavanie stavov sa využíva hook `useState`. Stav, ktoré ovplyvňujú viacero komponentov alebo je dobré ich spravovať spoločne, sú udržiavané pomocou Redux Toolkit. Celej aplikácii sú stavy poskytnuté pomocou `<Provider store=store>`. Aplikácia využíva slice pre tri oblasti. Hlavná je `canvas`, ďalšie sú `toolbar` a `style` na nastavovanie označeného nástroja, zvolenej farby, zvolenej hrúbky a podobne. Slice je vytvorený pomocou `createSlice` na uľahčenie písania reducerov. Najkomplexnejší je `canvasSlice`. Zdieľajú sa cez neho rôzne kontrolné hodnoty ako `isLoading`, `isDrawing`, `isDragging`, `isExporting`. Najdôležitejší je objekt `image`. Uchováva všetky nakreslené objekty, ktoré spadajú pod obrázok. Ukladá predošlé stavy objektov, pre vrátenie kroku späť. Každý `slice.js` súbor obsahuje aj selektory. Ich cieľom je sprístupniť stavy.

6.1.3 Komponenty

Aplikácia obsahuje väčšie množstvo komponentov. Uvediem preto len niektoré. Ako komponenty pre plátno, panel s nástrojmi, nástroje a upravovanie.

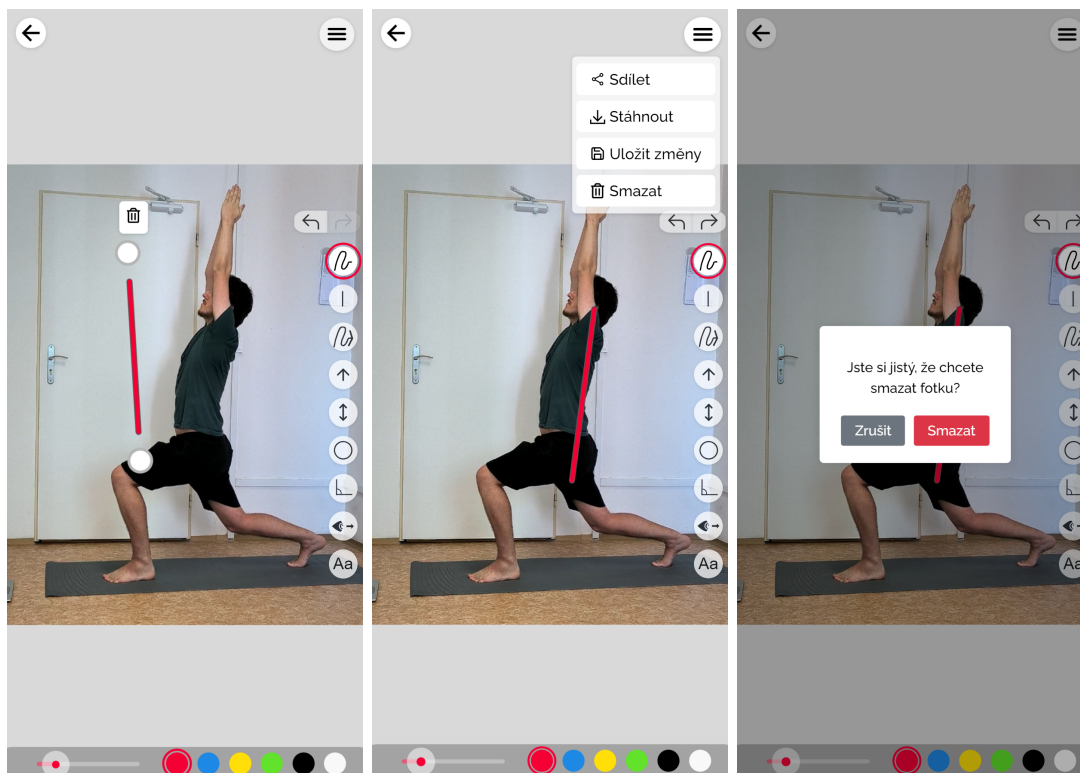


Obr. 6.1: Snímky prvých pohľadov pri otvorení aplikácie v rôznych štádiách vývoja. **Vľavo:** Pôvodná verzia, s jedným pohľadom a bez ukladania fotiek. **Vpravo:** Aktuálna verzia prvého pohľadu <Gallery/>. Pridanie galérie na zobrazenie predošlých označení umožňuje spätne opraviť užívateľovi chybu v komentároch. Rozšírenie o galériu zjednodušilo aj priebeh užívateľských testov.

Canvas

Komponent, ktorý spravuje <Stage/> a spracováva všetky udalosti na ňom. Vytvára nové objekty podľa stavu `selectedTool` a stavov v `styleSlice` a na základe typu vytvoreného objektu ho vykresľuje do plátna <Stage/>. Plátno som rozdelil do troch vrstiev. Prvej pre komponenty, ktoré nereagujú na dotykové udalosti ako pozadie plátna <PoseImage/> alebo <CommentArea/> na zobrazenie vysadených komentárov pod fotku pri exportovaní. Nasledovne, vrstva na kreslenie všetkých tvarov. Posledná na kreslenie hotspotov pre komentáre, aby neboli prekrývané.

Spôsob kreslenia sa vykonáva podľa zvoleného nástroja. Pokiaľ ide o komentár, tak sa spracuje pri `onTouchDown`, nastaví sa bod a tam to končí. Pri ostatných tvaroch a voľných kresbách sa vytvorí prvotný bod pri `onTouchDown` a nasledujúce sa pridávajú pri `onTouchMove`. Funkcia `updateFreehand` aplikuje nové body pri voľnom kreslení a funkcie `updateLine`,



Obr. 6.2: Snímky nástroja v pohľade <Tool/> **Vľavo:** Označený objekt je možné modifikovať farbou, hrúbkou, posunom alebo sa môžu zmazať. **Uprostred:** Možnosti čo s označenou fotkou vykonať. **Vpravo:** Modálne okno na potvrdenie vykonania akcie

updateAngle, updateCircle na tvorenie tvarov cez kliknutie a ťahanie. Z pozorovaní som zistil, že počas kresby, účastníci testov kreslili tvary voľnou rukou pomalším pohybom. To spôsobovalo zhluk bodov a škaredý efekt kresby, pri kresbe šípok ovplyvňoval aj smer šípky. Preto som implementoval jednoduchú funkciu na filtrovanie kresieb týmto nástrojom, SmoothLine na výpise 6.2.

```

<Layer>
  {[...Object.values(adjustedObjects), newObject].map((object) => {
    if (object.type === "comment") {
      return (
        <Comment
          key={object.id}
          object={object}
          isDraggable={!viewOnly && selectedObjectId === object.id}
          isSelected={selectedObjectId === object.id}
          onSelect={() => handleSelect(object)}
          viewOnly={viewOnly}
        />
      );
    } else {
      return null;
    }
  })}
</Layer>

```

Výpis 6.1: Příklad ako sa v Reacte mapujú objekty na komponenty do plátna vo vrstve Layer.

```

function smoothLine({ line, step, threshold = 100 }) {
  const points = line.points;

  // Step must be an even number
  if (line.points.length <= threshold || step % 2 !== 0) return line;

  // Values at index i and i + 1 are x and y
  const filteredLine = {
    ...line,
    points: points.filter((_, i) => i % step === 0 || i % step === 1),
  };

  return filteredLine;
}

```

Výpis 6.2: Funkcia na vyfiltrovanie bodov v čiarach, pre zabránenie zhlukov. Ponecháva iba prvky na indexe, ktoré sú periódou kroku a prvok hneď po ňom.

Panel s nástrojmi

Komponent na zobrazenie všetkých dostupných nástrojov s ktorými sa dá pracovať na plátne. Sleduje stavy ostatných komponentov pomocou `useSelector`. Na základe stavov `isDragging` a `isDrawing` schováva tlačidlá počas interakcie s objektami plátna. Panel tvoria tlačidlá `ToolButton`. Tlačidlá nastavujú pomocou `dispatch(selectTool(type))` vybraný nástroj. Označené tlačidlo využíva aktuálnu zvolenú farbu, s ktorou sa kreslí.

Nástroje

Celkovo je nástrojov v panely deväť, ale komponentov je len osem. Obojstranná šípka je len modifikácia obyčajnej šípky pomocou prop v `<Arrow doubleSided=true/>`. Medzi komponenty nástrojov patria: `Angle`, `Arrow`, `Circle`, `Comment`, `FreeHand`, `FreeHandArrow`, `Line`, `FreeHand`.

```
<LineKonva
  id={line.id}
  points={points}
  opacity={isOpacityLowered ? LOWERED_ALPHA : 1}
  stroke={line.color}
  strokeWidth={line.strokeWidth}
  hitStrokeWidth={line.strokeWidth * HIT_DETECTION_MULTIPLIER}
  lineCap="round"
  onTap={onSelect}
  onClick={onSelect}
/>
```

Výpis 6.3: Na vloženie tvaru do plátna sa používajú tvary knižnice React Konva. Nastavenia sa poskutujú pomocou props, ktoré spravuje každý komponent nástroja tejto práce.

Paleta s farbami a nastavením hrúbky

Rovnako ako panel s nástrojmi aj táto paleta sleduje stav aplikácie a v prípade posúvania alebo kreslenia sa schováva. Pri označení objektu sa paleta vysunie vyššie na poukázanie možnosti úpravy už umiestnených objektov. Skladá sa z dvoch typov komponentov. `StrokeWidthSlider` a `ColorButton`.

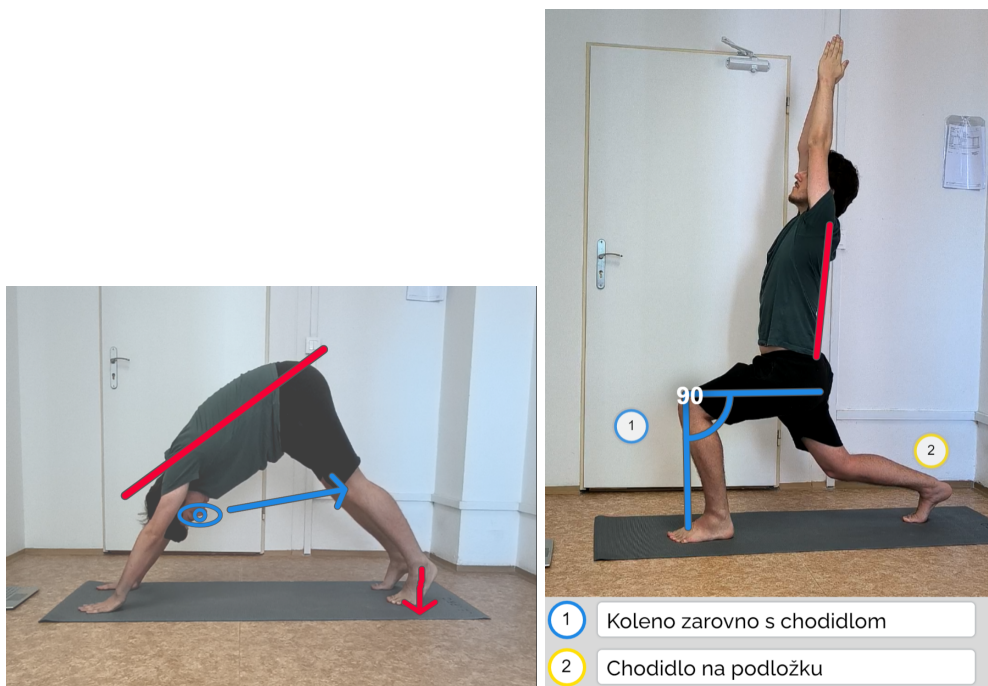
`StrokeWidthSlider` ma nastavenú minimálnu, maximálnu a aktuálnu hodnotu, podľa ktorých sa vykresluje. `ColorButton` pri kliknutí prejde do aktívneho stavu a nastaví aktuálnu farbu v plátna na svoju hodnotu. Tvarom sa mení hodnota farbu a hrúbky pomocou `useAdjustColorAndWidth`. Ten sleduje a porovnáva hodnoty objektu plátna s hodnotami v palete. Pri zmenách aktualizuje zvolený objekt.

6.1.4 Vstup a výstup

Aplikácia poskytuje nahrávanie fotografií niekoľkými spôsobmi. Jednoduchší spôsob je prístupný v galérii. Na vrchu je vložený komponent `<Upload/>`. Jeho úlohou je prijať súbor a transformovať ho pomocou `URL.createObjectURL` na súbor dočasne dostupný v prehliadači pomocou vygenerovanej URL.

Výstupom aplikácie je okomentovaná fotografia dostupná na stiahnutie do zariadenia alebo zdieľaná pomocou `navigator` z `Web Share API`¹ do inej aplikácie.

¹[Web Share API](#) – Zdieľanie súborov do iných aplikácií.



Obr. 6.3: Výstup aplikácie, ktorý sa môže poslať cvičencovi. **Vľavo:** V prípade, keď nie je použitý textový komentár, nie je k fotke pridaná spodná lišta s popisom. **Vpravo:** Užívateľ počas komentovania pridal text, preto sa pod fotkou pridá priestor na oddelené komentáre.

6.1.5 Nasadenie

Implementácia je dostupná na odkaze [Poses](#) pomocou Github Pages². Github Pages umožňujú jednoduchý a bezplatný hosting, poskytovaný cez zabezpečenie HTTPS, jedna z podmienok pre inštaláciu PWA na zariadení. Aplikácia má implementovaný backend na ukladanie fotografií a vložených anotácií. Rozhodol som sa tomu venovať čas z dvoch dôvodov. Prvý bol výmena fotiek pred testom bez potreby manuálneho menenia fotografií v kóde. Druhý dôvod bol pre uloženie upravených fotografií počas testovania a nasledovné pozieranie po testoch. Databázu aj úložisko poskytuje Supabase³. Funkcie na prácu s API Supabase v súbore `supaBaseAPI.js` sú vygenerované pomocou Supabase, vyžadovali len malé úpravy.

6.2 Zhodnotenie

Výsledkom je užívateľské rozhranie reprezentované formou interaktívnej progresívne webovej aplikácie. S určením pre komentovanie pozícií pomocou grafických nástrojov. Rozhranie vzniklo iteratívnym upravovaním, rozširovaním alebo odstraňovaním komponent. Tento prístup mi umožnil vidieť ako užívateľ používa prvky rozhrania a mohol som na základe pozorovania vykonávať zmeny.

Celkovo sa testovania implementovaného rozhrania zúčastnilo deväť inštruktorov, jeden osobný tréner a štyria účastníci boli mimo cieľovej skupiny. Testy boli zamerané na kvalitu, nie kvantitu. Nevyhodnocoval som úspešnosť, poznačoval som si správanie užívateľov počas

²[Github Pages](#) – nástroj na sprístupnenie aplikácie cez internet.

³[Supabase](#) – Ukladanie dát do tabuliek a fotografií do úložiska pre webové aplikácie.

skúšania aplikácie, ich pripomienky, myšlienky a nápady. Medzi testami som vykonával úpravy. Medzi zaujímavé zistenia počas testov patrili:

- Schovávanie spodnej palety – Paleta s farbami a nastavením hrúbky sa pri zrušení označenia schovávala a pri označení objektu alebo nástroja objavila. Účelom schovávanía bolo uvoľniť priestor na obrazovke pri približovaní a posúvaní po plátne. Zároveň sa označením objektu zas objavila. Naznačilo to možnosť úpravy pomocou palety. Paleta sa už iba vysunie kúsok vyššie, nemizne pri ukončení označenia.
- Posunuté kotvy od objektu – Síce nie priamo problém pre použiteľnosť väčšiny, ide skôr o obmedzenie. Účastníčka testu mala dlhšie nechty a nimi si prekrývala priestor od kotvy po začiatok objektu. Priestor na interakciu s kotvami bol nastavený dostatočne, aby naznačila čo chcela. U ostatných boli na tento prístup pozitívne ohlasy.
- Schovávanie paliet a tlačidiel počas procesu kreslenia účastníci často nevnímali.
- Interakcia s kotvami – Účastník skúšal upraviť veľkosť kruhu pomocou pinch gesta, pretože si nevšimol kotvu. Osoby, ktoré neprehliadli túto kotvu ju použili. Druhým zaujímavým pozorovaním bolo, že pri nástroji uhlu, kde sú kotvy od začiatku bližšie k sebe, účastníci testu použili dva prsty. Jeden prst na každú kotvu a nastavovali uhol pomocou viacerých dotykov. Táto funkcia bola podporovaná prototypom, avšak nebola priamo úmyselná. Tento prístup je naďalej v rozhraní povolený.
- Niektorí účastníci vôbec nevyužili priblíženie a oddialenie obrazovky počas celého priebehu testu.
- U niektorých som ukázal aplikáciu vo `viewOnly` móde na začiatku. Ide o zobrazenie bez paliet. Zaujímalo ma či by zistili, že hotspoty obsahujú dodatočné schované komentáre pri rozkliknutí. Najprv som použil rovnaké hotspoty ako v móde na kreslenie. Nenapadlo im kliknúť. Pri ďalšom testovaní som skúsil pridať pulzujúci efekt pomocou rastúcich tieňov objektu. Tieto už upútali pozornosť. Ku koncu však pri móde na editovanie fungovali hotspoty tak ako mali aj bez animácie. Otvorenie okna na písanie text po vložení komentára na plátno stačilo na spojenie tohto bodu na plátne s otvorením komentára.
- Kreslenie kruhu od stredu von – tento prístup prišiel viacerým veľmi zvláštny. Síce mohol vytvoriť kruh presne od bodu kam chceli. Mnohí to však skúšali bežným spôsobom od ľavej spodnej časti diagonálne hore doprava.
- Obmedzenie maximálneho oddialenia so zarovnaním na stred poskytlo užívateľom lepší pohľad než keby bolo plne odomknuté a užívateľ by si musel manuálne hľadať stred.
- Účastníkom testu niekedy trvalo pár pokusov nastaviť uhol na zaokrúhlenú hodnotu.

Rozhranie je interaktívne a funkčné pre účel tejto práce. Založené na výskume s užívateľmi cieľovej skupiny. Výskum poskytol náhľad do aktivít inštruktorov a poukázal aj na spôsob akým by značili pozície. Pre budúcnosť by mohla byť demonštračná aplikácia prerobená do natívnej aplikácie. Počas testov v posledných fázach boli nájdené nekompatibility knihovne na prácu s plátnom s niektorými prehliadačmi. Príkladom je Brave. Optimálne prostredie je prehliadač Chrome. Počas testov som počul niekoľko rád a nápadov kam by mohlo rozhranie pokračovať. Inštruktorka, ktorá robí okrem jogy aj pilates by si vedela

predstaviť podporu ďalších športov. Mohla by byť rozšírená na zapojenie videa a zvuku. Počas rozhovoru som sa dozvedel o novom type jogy, bodybalance. Ide o jogu, spojenú s pilates a Tai chi a zapojením hudby. Zostavy majú určité časovania spojené s hudbou a vyžadujú sledovanie osobou na kontrolovanie správneho cvičenia. Ďalšia možnosť je zapojenie užívateľského rozhrania na platformu, kde inštruktori poskytujú cvičencom videá. Na konci videa by mohlo byť urobené rozhranie pre navrhnutie užívateľovi urobiť si fotku, poslať ju do danej aplikácie a časom by sa osobe vrátili rady.

Kapitola 7

Záver

Cieľom tejto bakalárskej práce bolo vytvoriť užívateľské rozhranie pre komentovanie športových pozícií, so zameraním na jogové pozície a preskúmať jeho použiteľnosť.

Zistil som, akými formami sa učí, aké pozície sa vykonávajú počas cvikov, ako prebieha komunikácia medzi inštruktormi a ich cvičencami. Tento výskum som vykonával pomocou účasti na hodine v štúdiu aj online, rozhovormi s inštruktormi a osobnými trénermi, ale aj s literatúrou o joge. Vytvoril som fotky na komentovanie. Z nich som vybral 20 a vytlačené sady som rozdal inštruktorm s úlohou komentovania. Táto úloha mi pomohla s výberom tvarov, farieb a celkovo mi ukázala na čo sa inštruktori zameriavajú.

Zaoberal som sa aj podporou jogy pomocou aplikácií. Väčšina aplikácií k joge sa zameriavala na domáce cvičenie alebo plány na chudnutie, ale našiel som aj aplikácie na anotáciu fotiek. Tieto aplikácie mohli napodobniť značenia papierovej formy výskumu. Niektoré aplikácie sa zameriavali na komentovanie športov, ako CoachNow (na obrázku 4.5).

Z poznatkov som vytvoril prototypy s účelom testovania prvkov rozhrania, ktoré by boli súčasťou finálneho rozhrania. Pri tomto testovaní som sa držal kvalitatívneho prístupu „Do-it-yourself“ a využíval som predovšetkým existujúce aplikácie, interaktívne prvky tvorené vo Figma alebo aj čiastočné implementácie v Reacte, pokiaľ bola potrebná vyššia forma interakcie akú ponúkala Figma. Časti rozhrania som spojil do progresívnej webovej aplikácie, vyvinutej pomocou Reactu, na vytvorenie uceleného užívateľského rozhrania. Implementovanú aplikáciu som testoval a iteratívne vylepšoval.

Počas testovaní som si všimol rôzne prístupy k používaniu aplikácií. Najväčší rozdiel bol pri kreslení. Napriek tomu, že väčšina účastníkov prišla na prístup označenia nástroja a kreslenia pomocou priloženia prstu a ťahania alebo voľného kreslenia, boli účastníci, ktorí skúšali značiť pomocou ťahania a pustení. Schovávanie tlačidiel počas kreslenia si viacerí nevšimli, čo je pozitívny výsledok. Rovnako bol pozitívny prístup k riešeniu komentárov pomocou hotspotov.

Existuje veľa možností ako by mohla práca ďalej pokračovať. Aktuálne je demonštračná aplikácia zameraná hlavne na UI a UX, má zakomponované iba prvky backendu, na uľahčenie testovania. Mohla by sa rozšíriť, umožňovať sledovanie pokroku cvičencov medzi prvou, predošlou a aktuálnou formou počas cviku. Aplikovať lepší algoritmus na vyhladzovanie čiar. Počas užívateľských testovaní mi jeden účastník povedal, že by si túto funkcionality vedel predstaviť na platforme na prehrávanie cvikov, kde by sa na konci cviku poskytla sledujúcim možnosť poslať fotografiu ako vykonávajú precvičené pozície a inštruktor by im poskytol rady.

Literatúra

- [1] ABRAMOV, D. et al. *Getting Started with Redux*. 1. vyd. Redux.js.org, marec 2024. Dostupné z: <https://redux.js.org/introduction/getting-started>.
- [2] ANANT. React: Styled-components: A Modern Approach. *Medium*. 2023. Dostupné z: <https://medium.com/@anant3104/react-styled-components-a-modern-approach-084438672f94>.
- [3] DILARAULUTURHAN. New Generation Frontend Development Experience – Vite. *Medium*. December 2023. Dostupné z: <https://medium.com/bursa-bili%C5%9Fim-toplulu%C4%9Fu/new-generation-frontend-development-experience-vite-a4c7b264a157>.
- [4] EMADAMERHO ATORI, N. Beginner’s Guide to Svelte and SvelteKit. *Tech stack*. October 2023. Dostupné z: <https://prismic.io/blog/svelte-and-sveltekit>.
- [5] IYENGAR, B. K. S. *Light on Yoga*. 2. vyd. Schocken Books, 1977. ISBN 0-8052-1031-8.
- [6] JENIFER TIDWELL, A. V. *Designing Interfaces: Patterns for Effective Interaction Design*. 3. vyd. O’Reilly Media, Inc., 2020. ISBN 978-1-492-05196-1.
- [7] KHAN, H. Simplifying Progressive Web App (PWA) Development with Vite: A Beginner’s Guide. *Dev.to*. July 2023. Dostupné z: <https://dev.to/hamdankhan364/simplifying-progressive-web-app-pwa-development-with-vite-a-beginners-guide-38cf>.
- [8] KRUG, S. *Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems*. Pearson Education, 2009. ISBN 0321657292.
- [9] KRUG, S. *Don’t Make Me think: Revisited*. 2. vyd. Pearson, 2014. ISBN 9780321965516.
- [10] MARSH, J. *UX for Beginners: A Crash Course in 100 Short Lessons*. O’Reilly Media, 2016. ISBN 978-1491912683.
- [11] *Learn React*. React. Dostupné z: <https://react.dev/learn>.
- [12] SCHULTZ, L. *Ashtanga Yoga*. Nauli Press, 2000. ISBN 978-0-12-818202-4.
- [13] STAIANO, F. *Designing and Prototyping Interfaces with Figma*. 2. vyd. Packt Publishing Ltd, 2023. ISBN 978-1835464601.

- [14] UPENDRAPRASADMAHTO. React vs React Native: Key Differences. *Medium*. November 2023. Dostupné z: <https://medium.com/@upendraprasadmahto652/react-vs-react-native-key-differences-80237d4afc6f>.
- [15] WOHLWEND, B. A deep dive into KonvaJS. *Medium*. 2023. Dostupné z: <https://medium.com/@brandon93.w/a-deep-dive-into-konvajs-c5b88a161679>.