



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

**SDÍLENÝ PRACOVNÍ PROSTOR PRO SPOLUPRÁCI
ZÁKAZNÍKA S NÁVRHÁŘEM**

SHARED WORKSPACE FOR COLLABORATION OF A CUSTOMER AND A DESIGNER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ZDENĚK STUDENÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2017

Abstrakt

Práce popisuje návrh a vytvoření uživatelského rozhraní pro spolupráci klienta s designérem, které bude motivovat ke kontinuální spolupráci namísto jednorázových úkolů. Pro motivaci klienta je použitý koncept gamifikace, který se zabývá aplikováním herních mechanik do produktivních činností. Rozhraní je realizováno formou webové aplikace, jež si klade za cíl vytvořit u klienta návyk k jejímu pravidelnému používání a poté jej vybízí k dalšímu vývoji projektu. Provedené uživatelské testování ukázalo, že aplikace pomáhá lépe pochopit proces práce na projektu a úspěšně vyzývá klienta k dalšímu vývoji projektu.

Abstract

The thesis describes the design and creation of user interface made for cooperation of the client and the designer. This UI should motivate to keep the cooperation between the two parties continuous. It uses a concept of gamification which motivates the client. Gamification applies game mechanics into productive activities. The interface is realised in the form of web application aiming to create a habit of using the application on regular basis. Additionally, the application invokes further project development. Usability testing provided data suggesting that the application helps with better understanding of the working process of the project and encourages the client to develop the project further.

Klíčová slova

Uživatelské rozhraní, gamifikace, motivace uživatele, klient, designér, spolupráce.

Keywords

User interface, gamification, user motivation, client, designer, cooperation.

Citace

STUDENÝ, Zdeněk. *Sdílený pracovní prostor pro spolupráci zákazníka s návrhářem*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vítězslav Beran, Ph.D.

Sdílený pracovní prostor pro spolupráci zákazníka s návrhářem

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Vítězslava Berana, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Zdeněk Studený
16. května 2017

Poděkování

Tímto bych rád poděkoval mému vedoucímu bakalářské práce, panu Ing. Vítězslavu Beranovi, Ph.D za odborné rady a konzultace této práce. Také bych rád poděkoval Haně Jančiové a Romaně Guryčové za pomoc při formulaci myšlenek a korektuře práce.

Obsah

1 Úvod	2
2 Uživatelská rozhraní s použitím gamifikace	3
2.1 Jak dostupná uživatelská rozhraní motivují uživatele	3
2.2 Návrh uživatelského rozhraní	6
2.3 Gamifikací k vytvoření návyků uživatelů	9
2.4 Výběr technologií pro realizaci prototypu	16
3 Návrh webového rozhraní motivující uživatele	19
3.1 Proces spolupráce designéra s klientem a cíl projektu	19
3.2 Příběhem do cíle	22
3.3 Přidání motivačních prvků	23
3.4 Tvorba wireframů	25
4 Vývoj webové aplikace a testování	30
4.1 Vývoj webové aplikace S3PS	30
4.2 Uživatelské testování aplikace S3PS	34
4.3 Rozbor prvků na základě testování	36
4.4 Návrhy nových prvků	40
5 Závěr	44
Literatura	45
Přílohy	47
A Obsah přiloženého DVD	48
B Plakát	49

Kapitola 1

Úvod

Tvorba on-line projektů, jako jsou webové stránky nebo marketingové kampaně, bývá často považována menšími firmami jako jednorázová činnost s jasným cílem. On-line prostředí však dovoluje projekty neustále měnit a vyvíjet, to především na základě zpětné vazby a dat posbíraných z analytik. Důsledkem postupného vývoje a změn tak vznikají více optimalizované projekty, které lépe splňují definované cíle. To je výhodné jak pro zákazníky, kteří tak zvýší své tržby, tak pro designéry, kterým to zajistí stabilní přísun práce a dlouhodobě lepší výsledky z projektů.

Cílem této práce je navrhnout on-line pracovní prostor, který bude motivovat a podporovat dlouhodobou spolupráci a optimalizaci projektu. Celý pracovní prostor bude realizován formou webové aplikace, která bude sloužit také ke spolupráci zákazníka a designéra od samotného počátku společného projektu. Aplikace jako taková by tedy měla ulehčit společnou komunikaci a vytvořit návyk na její pravidelné užívání, které povede k dlouhodobé spolupráci.

První kapitola poskytne teoretický základ, který představí uživatelské rozhraní, gamifikaci, webové technologie a vztahy mezi zákazníky a designéry. V druhé kapitole je popsán návrh řešení, k jehož dosažení bylo třeba definovat motivace zákazníků, aplikovat gamifikační metody a vytvořit příběh, který provede uživatele aplikací. Nejprve bylo uživatelské rozhraní realizováno pomocí wireframů a později byla aplikována grafika. Ve třetí kapitole je popsána tvorba aplikace v React s použitím knihovny material-ui, aby následně mohla být testována a jednotlivé prvky poté vyhodnoceny.

Kapitola 2

Uživatelská rozhraní s použitím gamifikace

Tato kapitola vysvětluje, jak může správně navržené uživatelské rozhraní motivovat uživatele k pravidelnému používání a k vybudování dlouhodobé spolupráce. Nejprve jsou z pohledu motivace rozebrány světově používaná uživatelská rozhraní a je ukázáno, jaké prvky pro motivaci uživatele používají. Zaměřuji se zde také na to, jak jednotlivá rozhraní ulehčují práci na projektu.

Dále vysvětluji techniky dobrého návrhu uživatelského rozhraní, které bude pro uživatele pochopitelné, a tak nebude muset přemýšlet nad tím, jak jej používat. Jsou zde popsány principy, kterých se má uživatel při návrhu držet, a jak nad návrhem přemýšlet.

V návaznosti na návrh rozhraní se zabývám motivací uživatelů skrze gamifikaci. Z obsáhlých technik gamifikace jsem zde rozepsal především ty, které jsou v práci použity. Jsou zde představeny hlavní motivace, které vedou uživatele k akci, a vysvětleno jak s nimi pracovat v uživatelském rozhraní. Dále je zde popsána důležitost příběhu a emocí při komunikaci s uživatelem.

Závěr kapitoly se zabývá výběrem vhodné technologie pro vytvoření prototypu. Vybíral jsem ze dvou aktuálně nejpoužívanějších javascriptových knihoven a frameworků, což jsou React a AngularJS. Nejprve jsou zde popsány požadavky a následně rozebrány klady a zápory obou.

2.1 Jak dostupná uživatelská rozhraní motivují uživatele

Na trhu je spousta aplikací pro řízení projektů, které se zaměřují na zpřehlednění práce v týmu a dobrou organizaci jak lidí tak i výstupů, které během práce vznikají. Tyto aplikace také používají techniky, které motivují k plnění termínů a plánování úkolů. To ve výsledku vede k větší produktivitě týmu a přehledu o rozdělení práce v týmu a jejím progresu. Takto navržená aplikace je vhodná pro pravidelnou denní aktivitu interního týmu, nicméně není uzpůsobena pro klienta, který do ní bude nahlížet pouze několikrát do týdne. Taková aplikace pro něj poskytuje příliš mnoho detailů, čímž je zbytečně složitá.

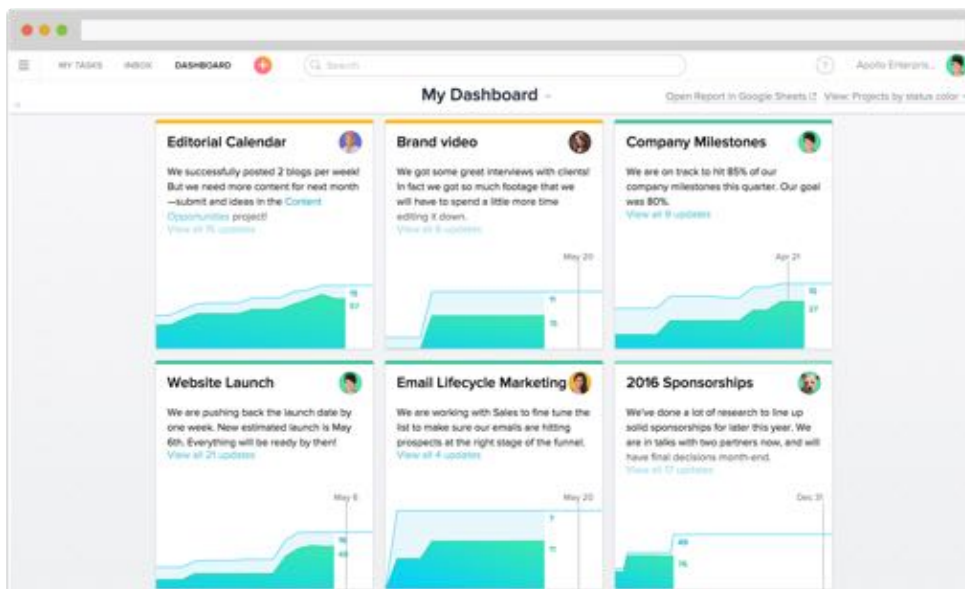
Existují také aplikace, které se zaměřují na spolupráci klienta se zákazníkem. Většinu komunikace na projektu tak přesouvají na jedno místo, kde zákazník vidí jak úkoly, tak hotové výstupy, nebo například počet hodin strávených na projektu a vystavené faktury. Tyto aplikace jsou pro spolupráci klient a designér výborné, zaměřují se však především na přehlednost projektu a dostupnost všech potřebných materiálů. Mým záměrem je na-

vrhnout rozšíření, které umožní, aby byl klient do projektu aktivněji zapojený, a měl zájem o kontinuální spolupráci a údržbu daného projektu v závislosti na datech získaných z provozu.

Asana

Tento nástroj, specializující se na organizaci práce, je založený na jednotlivých úkolech a sledování průběhu projektu. Samotné jádro aplikace je tedy velmi přímočaře zaměřeno na úkoly, je však možné jej upravit o spoustu rozšíření. Jedná se například o priority úkolů, skupiny, chaty a propojení s dalšími aplikacemi. To z Asany dělá výborný nástroj ke správě projektu.

Jak již bylo zmíněno, Asana pracuje také se sledováním průběhu projektu (lze vidět na obrázku 2.1) a nabízí možnost sledovat, kolik se který den splnilo úkolů, čímž motivuje uživatele k větší aktivitě. Uživatelé při každém příchodu do aplikace vidí graf splněných úkolů. Čím víc jich daný den udělají, tím strměji graf stoupá, což motivuje k tomu být alespoň natolik produktivní jako včera. To je skvělá motivace pro práci na projektu. Co zde chybí, je možnost plánovat do budoucna a motivovat klienta k delší spolupráci. To musí zajistit designér formou schůzek a další komunikace.



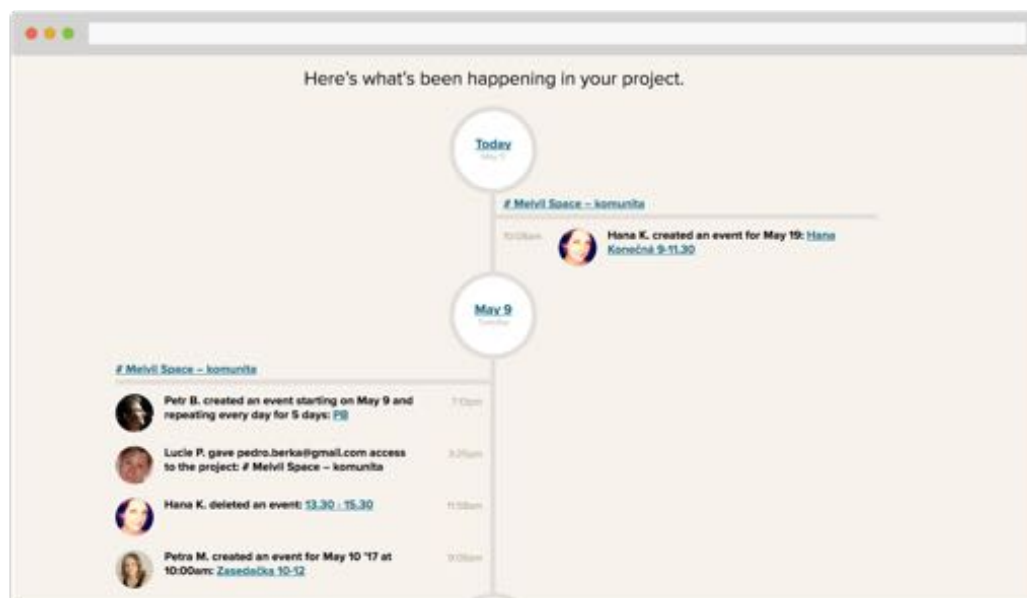
Obrázek 2.1: Při příchodu do Asany je vidět kolik úkolů jste již splnili. Jestliže budete dnes méně produktivní, na grafu to budě vidět.

Basecamp

Basecamp je webová aplikace pro organizaci týmu a práce. Zaměřuje se na veškerou komunikaci a události, které se během projektu dějí. Díky tomu tak tým nemusí používat několik různých aplikací a může mít konverzace a dokumenty na jednom místě. Je to skvělá aplikace pokud chcete mít přehled nad projektem, vědět kdo má jakou úlohu, jaké jsou termíny a znát veškeré výstupy.

Právě přehled nad soubory a aktivitami je zde ukázkový a v bodě, kdy si na něj uživatel zvykne, stane se pro něj Basecamp zásadní aplikací pro organizaci projektu. V motivaci

uživatelé k práci na projektu však Basecamp už není tolik zajímavý. Spoléhá především na zájem uživatelů dokončit projekt a na skupinový nátlak, kdy uživatel vidí, jak ostatní pracují, a sám nechce být ten, co projekt sabotuje. Basecamp se nejspíš snažil postupně přidávat i některé další motivační prvky, ale zařadil je spíše mimo hlavní jádro služby, viz například časová osa na obrázku 2.2. Celé rozhraní také působí hodně pracovně a ne příliš přátelsky. Jsou zde jednoduše všechny prvky, které potřebujete k práci na projektu a žádné zbytečné rozptylování.

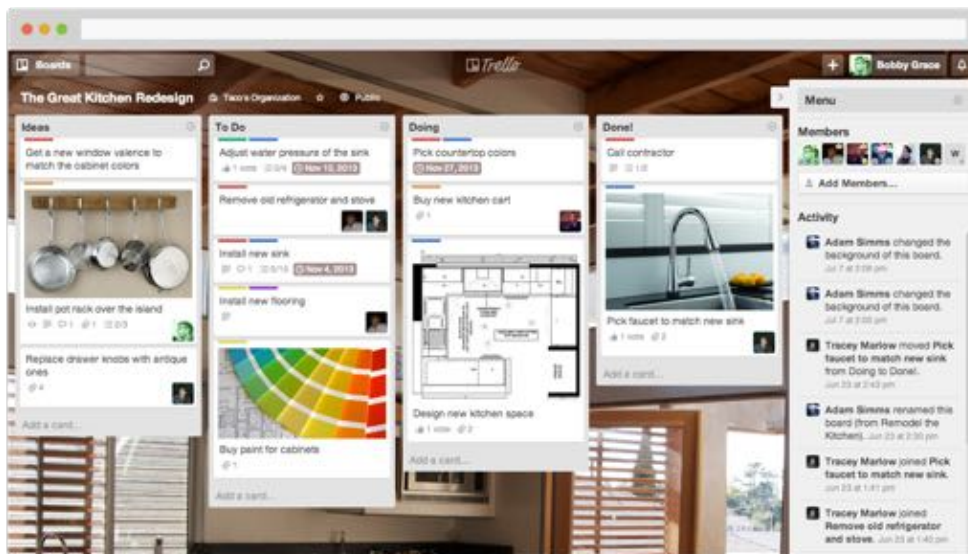


Obrázek 2.2: Stránka *Progress* ukazuje v Basecampu pokrok jednotlivých členů, přičemž v členech vyvolává potřebu být aktivní. Nedostatkem je, že s touto stránkou Basecamp nijak aktivně nepracuje. Při práci na projektu se na ni nedostanete a musíte se k ní sami proklikat mimo klasický pracovní cyklus.

Trello

Trello je nástroj, který nabízí velkou spoustu užití především díky své vizuální podobě. Základním prvkem Trelly jsou nástěnky, na kterých jsou potom jednotlivé sloupce a karty. Uživatel si jednotlivé karty přesouvá mezi sloupci a tím si projekt organizuje. Jak se Trello bude používat je tedy plně na uživateli. Standardním řešením může být vytvoření sloupců *To-do*, *Doing*, *Done*. Je jednoduché vytvořit si další sloupec specifický pro daný projekt. Tím mohou být například *Nápady* a *doporučení*, kde v uživateli budíte kreativitu a ukazujete mu možné způsoby jak v projektu pokračovat. Trello tak umožňuje při dobrém nastavení pracovat s každým projektem odlišně. Pro dlouhodobé plánování je to velmi cenné a je zde jednoduché zaznamenat všechny nápady, které je později možné v projektu využít. Rozložení sloupců a karet lze vidět na obrázku 2.3

Co se týče motivace, Trello z velké části spoléhá na uživatele a staví se do role nástroje, který jim má pomoci neztratit se v projektu. I přesto nabízí možnost plánovat termíny úkolů a nadále s nimi poté pracovat. Například mění barvy úkolů a upozorňuje uživatele na blížící se termín. V pravém sloupci jsou také vidět aktivity všech členů dané nástěnky, a tak je vidět, jak plní úkoly ostatní. U jednotlivých karet úkolů je možné přidávat seznamy



Obrázek 2.3: Hlavní výhodou Trelly je vizuální zobrazení, které se dá lehce upravit pro jakýkoliv projekt.

úkolů. Při plnění seznamu se uživatelům postupně plní progress bar a mohou tak sledovat svůj posun.

Rozšířením Trelly o přidání gamifikace se zabývá getbadges.io. Velkou nevýhodou ale je, že získáváte body na jiné stránce než je Trello, a tak najednou musíte používat dvě různé aplikace. Toto samo ještě nemusí být takový problém, pokud se rozšíření rozhodne používat tým. Nicméně, vysvětlovat tento přístup klientovi a nutit ho používat obě aplikace, je již velice nepraktické. Gamifikace by v rozhraní měla být přirozená a uživatele motivovat při běžném používání.

2.2 Návrh uživatelského rozhraní

Pro návrh uživatelského rozhraní je třeba pečlivě analyzovat svoje uživatele, jejich potřeby, navrhnout jejich prožitek s celou aplikací a poté výsledek vizualizovat do uživatelského rozhraní. Je proto třeba pracovat s vizuální hierarchií, vybrat barevnou paletu, navrhnout způsob dávání feedbacku uživateli a další principy, které dále v této kapitole popíši.

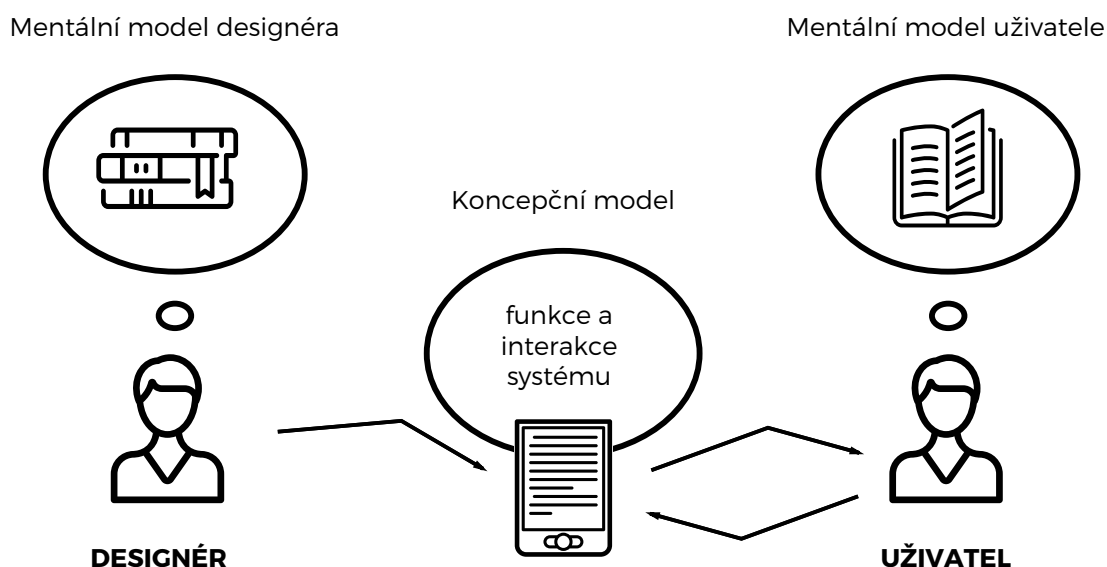
Uživatelské rozhraní a zkušenost

Anglicky také UI (User Interface) a UX (User Experience) jsou běžně rozšířené pojmy v designérské komunitě. Dá se říct, že UI je vlastně částí UX, kterou pro uživatele naplánoval designér. Pro návrh dobrého UI jsou zásadní informace od UX designéra, který celou **zkušenost** s aplikací naplánoval. Pocit z aplikace se neskládá pouze z uživatelského rozhraní, ale také z interakcí, pečlivě připravených textů a feedbacku. Zodpovědností UX designéra je celé rozhraní otestovat s uživateli. Teprve po testování se ukáže, jak rozhraní plní svůj cíl. UX designéři tedy musí dobře chápat své publikum a jeho potřeby. Teprve až jej pochopí, může vzniknout rozhraní, které bude plnit skutečné potřeby.

Koncepční a mentální model

Mentální model je reprezentace toho, jak osoba chápe rozhraní, s kterým interaguje. Koncepční model je poté aktuální model, který je osobě představen skrze design a uživatelské rozhraní produktu. Tento vztah je zobrazen na obrázku 2.4. Proč jsou tyto modely důležité? Pokud mezi mentálním a koncepčním modelem není shoda, bude pro uživatele těžší naučit se produkt používat. To se může stát z důvodů jako *nepochopení uživatelů, navržení produktu pouze pro jednu osobu a nepochopení ostatních* nebo *dovolení technologii, aby diktovala výsledný design*. [20]

Při nasazování nového produktu, jako je například tvorba čtečky knih namísto aktuálních knih, je důležité uživatele provést novým modelem a dovysvětlit mu jej. Pro ověření, zda uživatelé mají stejný koncepční a mentální model, je třeba aplikaci průběžně a aktivně testovat.



Obrázek 2.4: Mentální model je obraz systému, který má uživatel v hlavě. Ten se často liší od koncepčního modelu, který představuje skutečný obraz toho jak systém funguje.

Poznání publika

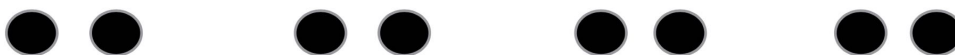
Největším překvapením, že poznání svých uživatelů před tvorbou rozhraní, je zásadní. Je třeba definovat si, kdo bude aplikaci používat, několik takových lidí potkat a zjistit jejich potřeby. Zde je extrémně nebezpečné dělat předpoklady o lidech bez konzultace s nimi. Mohou tak vznikat domněnky a vymyšlené problémy, což vede ke vzniku rozhraní, která neřeší reálné problémy uživatelů. Takový přístup pak způsobí ztracené měsíce práce.

Jednoduchost a konzistence

Klíčem k dobrému návrhu rozhraní je jeho jednoduchost. A jednoduchostí se nemyslí malý počet prvků. Jednoduchostí je myšleno, že uživateli bude jasné, co má dělat a nebude nad tím muset přemýšlet. Nebude třeba tvořit manuál, jak rozhraní používat, protože to bude jednoduché.

Používáním běžných prvků v rozhraní se uživatelé cítí pohodlněji a jsou schopni v něm rychleji pracovat. Při používání vzorců v designu i grafice skrze rozhraní, se uživatelé postupně stávají efektivnějšími v jeho používání. Pokud se uživatel jednou naučí jak úkol provádět, měl by to být schopen zopakovat i u dalšího úkolu. [19]

Lidské oči a mozek se automaticky snaží tvořit vzorce. A to i tam kde nejsou. Poznávání vzorců nám pomáhá činit rychlá rozhodnutí na základě spousty informací, které vstřebáváme každou vteřinu. Na obrázku 2.5 tak nejspíš vidíte čtyři dvojice teček, namísto osmi jednotlivých teček. Mozek totiž interpretuje množství místa mezi tečkami jako vzorec. [20]



Obrázek 2.5: Mozek chce vidět seskupení dvou teček jako vzorec, jak by však řada pokračovala dál není nikde dané.

Zapojení vizuální hierarchie

Nejdůležitější prvky rozhraní by měly být zvýrazněny, aby na ně uživatel upřel svou pozornost. Naštěstí design poskytuje spoustu způsobů, jak toho dosáhnout.

Nejjednodušší technikou je daný prvek udělat větší, a tím z něj udělat ústřední bod aplikace (stejně jako je těžké nedívat se v kině na plátno). Jiným způsobem může být použití bílého místa na zvýraznění důležitých částí rozhraní. Vytvořením prostoru okolo prvku se umocní jeho důležitost (stejně jako třeba vyniká brněnský orloj na Náměstí Svobody v Brně). Dalším způsobem poté může být porušení konzistence rozhraní. Pokud bude nějaký prvek vybočovat z určitého stylu, uživatel si toho všimne a bude zjišťovat, proč vybočuje a k čemu slouží. [20]

Použití barev a kontrastu

Výběr barev může být také dobrý nástroj pro nastavení vizuální hierarchie, výrazné barvy budou přitahovat pozornost na prvky, se kterými je třeba interagovat, nebo na prvky, kde je vyžadována pozornost uživatele (např. úkol není splněn v termínu a je třeba jej dokončit). Naproti tomu u prvků, které jsou stálé a opakující se, by se stejně tak měly opakovat barvy, a uživatel by si měl časem zvyknout na význam jejich užití. [8]

Zpětná vazba uživateli

Procesy v aplikaci jsou závislé na použité technologii, hardwaru a na interakcích uživatele. Někdy tak vykonání úkolu může trvat déle, než si uživatel představoval. A ať už se procesy v aplikaci provedou nebo ne, je dobré uživatele o nich informovat, umožní jim to zpřesnit svůj mentální model 2.2. Zároveň dostanou zpětnou vazbu na svou akci a to posílí jistotu v práci s aplikací.

V ideálním světě by každé rozhraní bylo natolik intuitivní a rychlé, že by uživatelé vůbec nemuseli přemýšlet, jestli se něco načítá, jestli něco zadali špatně, nebo zda se v pozadí něco počítá. Jenže hardware ještě není dost rychlý, internet dost dostupný a designéři dost předvídaví. Jednoduše si uživatelé vždy najdou cestu, jak rozhraní použít jinak, než bylo zamýšleno. Poskytnutím instantní zprávy, o tom co se děje, však můžete uživatele postupně učit, jak by rozhraní měli používat. [14]

Desing orientovaný na cíl

Design orientovaný na cíl je termín, který vysoce popularizoval Alan Cooper v jeho knize *The Inmates Are Running The Asylum: Why High-Tech Products Drive us Crazy and How to Restore the Sanity* publikované v roce 1999. Alan ho definuje jako design, pro který je řešení problému nejvyšší prioritou. Oproti ostatním přístupům vůbec nezohledňuje omezení technologie. Dnes se již tento přístup může zdát zřejmý, ve své podstatě je však aktuálnější, než kdy dřív.

Pro uplatnění přístupu je podle Alana třeba držet se pěti zásad:

1. **Designujte první, programujte až potom.** Uvažujte jak lidé interagují, a jak pro ně věci navrhnout (včetně vzhledu).
2. **Oddělte odpovědnost za design od programování.** Toto odkazuje na nezbytnost interakčního designéra, který si dokáže získat koncového uživatele, aniž by se zabýval technologickými možnostmi. Měl by věřit, že jeho developer zvládne vyřešit všechny technické potíže.
3. **Designér musí být odpovědný za kvalitu produktu a spokojenost uživatele.**
4. **Definujte si jednoho specifického uživatele produktu.** Je třeba si tak definovat jednoho vzorového uživatele, u kterého je třeba se ptát *Kde to bude používat? Kdo to je? Čeho chce dosáhnout?*
5. **Pracujte v týmech po dvou.** Interakční designér by neměl nikdy pracovat sám. V případě bakalářské práce je tak třeba práci aspoň s někým pravidelně konzultovat. Může to být profesor, kolega, nebo člen rodiny, ale je třeba vždy jít do hloubky. [5]

2.3 Gamifikací k vytvoření návyků uživatelů

Uživatelské rozhraní, které je přehledné a jednoduše zvládne veškerou potřebnou funkcionalitu, je více než dobrý základ. Je však možné jít ještě hlouběji a pokusit se vytvářet zdravé návyky při jeho používání. Takovým návykem může být například to, že uživatel bude aplikaci kontrolovat pravidelně, a že bude plnit své úkoly včas. Cílem je tedy **zapojení uživatele do celého procesu a vytvoření vazby na rozhraní.**

Zapojením uživatele se zabývá především *human-centered design*, který lze definovat jako proces, který považuje lidské potřeby a omezení za nejvyšší priority ve fázích návrhu a produkce. [7] *Human-centered design* se zaměřuje na budování vztahu mezi produktem (či službou) a uživatelem. Požaduje tedy dokonalé poznání svých uživatelů a analýzu jejich potřeb. Výsledný produkt tyto potřeby řeší a vytváří žádanou vazbu uživatele. Opačný přístup by mohl být zaměřený na přidání funkcionalit a aplikování nových technologií.

Obdobně se lidskou motivací zabývá také *human-focused design*, ten se zaměřuje na dodání zábavných a vtahujících elementů dříve obsažených ve hrách a jejich zasazení do reálného světa a produktivních aktivit. *Human-focused design* zohledňuje, že lidské emoce, ambice, pochybnosti a jejich důvody pro to, některé věci dělat a jiné ne. Tyto pocity a motivace jsou zohledněny při tvorbě celkového systému společně s jeho funkcionalitou. *Human-focused design* (neboli **gamifikace**), což je pojem vytvořen gamifikačním guru Yu-Kai Cho, se jeví jako dobrý přístup pro zapojení klientů k dlouhodobé spolupráci na projektu. [21]

Gamifikační koncept Octalysis

Gamifikace se zabývá aplikováním herních mechanik a technik pro vtáhnutí uživatele do děje za účelem dosažení definovaných cílů. Tak ji popisuje Yu-kai Cho ve své knize *Actionable Gamification: Beyond Points, Badges, and Leaderboards* [21]. Také tvrdí, že veškeré akce jsou způsobené jedním (nebo více) z osmi hnacích nábojů (anglicky Core Drive). To naznačuje, že pokud zde není žádný hnací náboj, není zde ani motivace konat.

Koncept Octalysis vznikl odvozením z osmiúhelníku (anglicky octagon), kde každá jedna strana reprezentuje jeden hnací náboj. Umístění hnacích nábojů na osmiúhelník není náhodné, ale vyjadřuje, jak na lidi různé motivace působí. Některé jsou například inspirativní a jiné zase zanechávají negativní pachůf. V Octalysis tak najdeme techniky povzbuzení na horních stranách osmiúhelníku a techniky manipulace zase na stranách spodních (Octalysis lze vidět na obrázku 2.6). [21]

Při zakreslení použitých hnacích nábojů v Octalysis tak designér získá přehled o tom, jaké hnací náboje se nejčastěji snaží vyvolat. Pokud se tato informace dále skombinuje s dobrou definicí cílové skupiny, navrhnuté rozhraní uživatele naprosto vtáhne. Například pro uživatele pracující ve finančnictví by bylo důležité aktivovat hnací náboj vlastnictví. U kreativců je zase důležité povzbudit a dávat rychlou zpětnou vazbu.

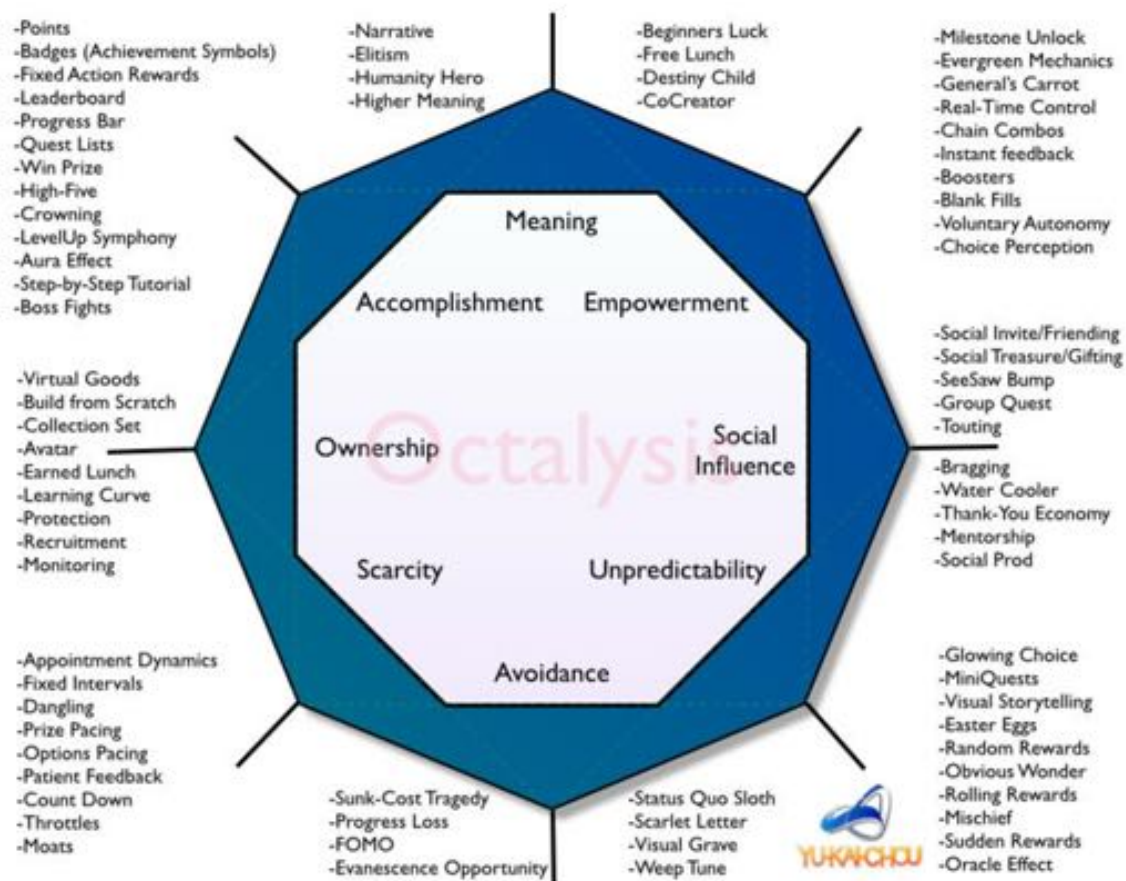
Hnací náboje, které jsem se rozhodl použít v rozhraní budu v této kapitole dále popisovat a ukazovat na příkladech použití. Ač pojem gamifikace je poměrně nový, podobné motivace popisují i designéři, kteří se gamifikací nezabývají. V práci tak dále ukazují, že byl koncept gamifikace používán dávno předtím, než získal svůj název.

Vize projektu a poslání

Velký cíl a možnost vidět k čemu jednotlivé malé úkoly vedou motivují lidi už po staletí. Díky tomu se lidé schází na protesty, díky tomu když Martin Luther King prohlásil že má sen, se sešlo tolik lidí ve správný čas na správném místě a to bez možnosti hromadné domluvy na internetu. Lidé si to navzájem řekli, protože věřili tomu, co on. Protože sdíleli stejnou vizi a poslání, rozhodli se cestovat i několik hodin, aby ji podpořili. Ze stejného důvodu funguje Wikipedie a je z ní dnes největší databáze lidského vědění, přestože se jiné společnosti snažily na tento úkol najmout vědce a platit přispěvatele. Wikipedie vznikla skupinou nadšenců, kteří ji tvořili zadarmo jen kvůli své silné vizi. Mít jasný cíl a poslání dokáže lidi nadchnout natolik, že vůbec neřeší vlastní zisk. [18]

Pro vyvolání takovéto motivace je třeba mít vizi a každý jednotlivý zúčastněný musí cítit, že i on je ten důležitý prvek, který pomáhá k naplnění této vize. Yu-kai Cho tuto motivaci zmiňuje jako hnací náboj epického poslání a staví ji na vrchol Octalysis jako nejvíce inspirující. Susan Weinschenk [20] ve své knize *100 věcí, které by měl každý designér vědět o lidech* zase zmiňuje, jak je důležité motivovat lidi skrze vnitřní motivaci a vlastní přesvědčení. Díky tomu budou chtít do projektu dlouhodobě přispívat a budou se cítit naplnění.

V kontextu rozhraní je třeba uvědomit si, proč celý projekt vzniká. Je nutno jít hlouběji. Nestačí vědět že vzniká nový web, protože ho klient chce. Vzniká nový web, protože klient chce ukázat celému internetu, kdo je a co dokáže. Poslání celého projektu by se v rozhraní mělo nést a neustále připomínat, tak aby se i menší úkoly plnily s jasnou vizí.



Obrázek 2.6: Octalysis a osm hnacích nábojů vedoucí k akci. Strany jsou rozděleny stejně jako hemisféry v lidském mozku. Na pravé straně tak najdeme hnací náboje spojené s kreativitou, sebevyjádřením a sociálním aspektem. Na levé potom hnací náboje spojené s logickým uvažováním, počítáním a vlastnictvím.

Cesta k cíli

Osobním rozvojem a cestě k cíli se věnuje Susan Weinschenk. Důležitost přikládá především možnosti pomoci lidem nastavit cíle a najít způsob jak sledovat jejich posun k těmto cílům. Samotná cesta za dosažením mistrovství je tedy to důležité a je třeba ji zaznamenávat a ukazovat uživateli. Právě pocit pohybu vpřed a překonávání překážek uživatele motivuje jít dál a nevzdat se po všem, co už dokázal. [20]

Ve hrách tento posun často bývá znázorněný sbíráním zkušeností a dosahování nových levelů. K sledování získaných zkušeností slouží takzvané *experience bary*, kde hráč může sledovat, jak se postupně plní po dosažení nového levelu. Za plnění úkolů pak hráči bývají odměněni bonusovými zkušenostmi a herní měnou či odznaky.

Při návrhu uživatelského rozhraní lze pracovat s obdobou *experience barů* v podobě *progress barů*, které ukazují, kolik úkolů už je hotových, a motivují uživatele naplnit bar na sto procent. Obdobně jako se ve hrách odměňuje za splnění úkoly, i v rozhraní je důležité uživatele pochválit za jejich plnění a darovat jim odměnu. Častou odměnou u gamifikovaných rozhraní bývají právě odznaky nebo body.

Podpora kreativity pro řešení překážek

Hnací náboj kreativity a zpětné vazby je zapojován, když jsou uživatelé v procesu tvorby, kdy neustále zkouší nové věci a různé kombinace. Lidé však nepotřebují pouze způsoby, jak vyjádřit svou kreativitu. Potřebují také vidět výsledek své kreativní činnosti, získat zpětnou vazbu a mít možnost vytvořit dále upravovat. Právě proto je hraní si s LEGEM a tvorba umění přirozeně zábavné. [21]

Při kreativní činnosti plně postačí vidět vytvořené dílo a v duchu si říct, jak jsme s ním spokojeni a kde by šlo vylepšit. U her jsou pak jasně dané úspěchy a herní mechanismy, které samy rozeznají, zda je vytvořené řešení dostatečně dobré, nebo je třeba na něm ještě zapracovat. U tvorby digitálních projektů je však potřeba získat feedback od cílového publika kampaní a webů, což můžou zajistit služby analytiky jako například Google Analytics a Hotjat. Nebo je možnost dostat feedback od zbylých členů týmu před samotným nasazením projektu. Je tedy nutno motivovat uživatele, aby navzájem svoje nápady konzultovali, a aby to dělali pravidelně, jelikož důležitá při podporování tohoto hnacího náboje je právě rychlá zpětná vazba.

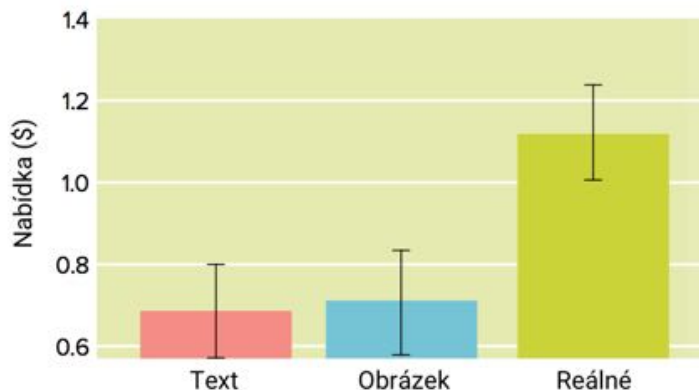
Pocit vlastnictví

Pocit vlastnictví je vyvolán, když uživatel cítí, že má nad něčím kontrolu. Když někdo cítí vlastnictví, chce jej postupně navyšovat. Jedná se o hlavní motivaci spojenou s touhou kumulovat bohatství a překvapivě obdobně dobře funguje při kumulaci virtuální měny v systému. Podobně je vnímána i časová investice. Když člověk stráví spoustu času upravováním svého profilu, automaticky na něj bude více hrdý a bude si dávat pozor na to, aby se ukazoval v dobrém světle. Hnací náboj vlastnictví je aktivován, když uživatel cítí vazbu k projektu, procesu a nebo organizaci. [21]

Tato motivace je obzvláště zajímavá v on-line prostředí. V roce 2010 Ben Bushong a jeho tým provedli experiment, aby porovnali kolik jsou lidé ochotni zaplatit za jimi známý produkt, pokud jim je představen textovou formou, obrázkem nebo pokud je reálně před nimi. Zúčastněným byly dány peníze a mohli si vybrat ze spousty svačinek (jako jsou brambůrky, sladkosti, tyčinky, atd.). Některé svačinky byly popsány pouze textově, u jiných byl obrázek a jiné byly reálně před účastníky experimentu. Účastníci poté měli určit, kolik jsou ochotni

za jednotlivé svačinky zaplatit. Rozdíly ceny jsou zobrazeny na obrázku 2.7 a ukazují, že rozdíl mezi opravdovou věcí a textem je skoro dvojnásobný.

Oproti reálnému světu nelze v rozhraní působit na všechny smysly, je zde proto motivace aktivovat jich co nejvíce a co možná nejsilněji. Proto je lepší používat obrázky namísto textů a animované vizualizace namísto obrázků, když to situace dovolí.



Obrázek 2.7: Graf ukazuje jak si lidé nacenili svačinky podle toho, zda byly popsány textově, ukázány na obrázku a nebo reálně před nimi. Čím více smyslů je věc schopná aktivovat, tím více si ji bude člověk cenit.

Jeden tým a každý maká

Společenskou sounáležitost využívá spousta on-line her v dnešní době. Pokud si během hraní vytvoříte s ostatními hráči přátelský vztah, začnete si plánovat společné akce ve hře a cítíte společenskou povinnost na ně dorazit. Obdobně, pokud si při práci na projektu s ostatními členy vytvoříte přátelský vztah, nebudete je chtít zklamat a některou část odbýt a nebo odevzdat pozdě. Tato motivace funguje také, pokud vidíte, jak ostatní pracují. Protože jste součástí týmu, začnete cítit potřebu se do práce na projektu zapojit.

V uživatelském rozhraní je tak dobré budovat místa, kde je možné s ostatními uživateli komunikovat a podporovat vznik přátelských vztahů. Při těchto komunikacích je třeba zároveň ukazovat avatara a nenechat komunikaci anonymní. Stejně tak důležité je ukazovat, na čem pracují ostatní, díky čemuž uživatel uvidí, že pokud nesplní svoje úkoly, bude na něj muset zbytek týmu čekat. Důležitost této motivaci přikládá také Yu-Kai Cho, který ji přikládá odpovědnost za velký úspěch sociálních sítí v dnešní době. [21]

Nepředvídatelnost, která udrží bdělost

Při nepředvídatelnosti je člověk neustále angažován, protože neví co se stane dál. Pokud něco nezapadá do klasických vzorců, mozek zařadí vyšší otáčky a začne se tomu věnovat. Jedná se pochopitelně o hnací náboj odpovědný za závislosti na gamblingu, ale je také přítomný v každém loterijním programu. Na vyšší úrovni také spousta lidí sleduje filmy a čte knihy právě kvůli tomuto hnacímu náboji. [21]

Tento princip již dříve popsali Daniel Kahneman. Při porušení klasických vzorců mozek vypíná autopilota a zpozorní kognitivní systém 1, ten nadále aktivuje analytický systém 2, který se nad problematikou dále zamýšlí a hledá řešení. [10] Vytvářet nepředví-

datelné chování v rozhraní tedy nemusí být vůbec žádané, nechci totiž, aby uživatel nad používáním rozhraní přemýšlel.

Nepředvídatelnost však lze aplikovat při obdarovávání uživatele za práci na projektu. Zde závislost podobná gamblingu nevadí, jelikož vede pouze k více splněným úkolům a tím k splnění cíle. Susan Weinschenk [20] zmiňuje, že klasické odměny je třeba neustále zvyšovat, aby uživatele uspokojily. Zatímco při darování odměn nepravidelně cítí uživatelé pořád stejné uspokojení, což je další výhodou použití nepředvídatelnosti.

Předejít negativním událostem

Motivace vyhnout se negativním událostem a nepromeškat ty pozitivní je velice silná a často nás nutí dělat věci, které se nám aktuálně dělat nechtějí. V menším měřítku může jít o ztrátu práce, nebo změnu něčeho chování. Ve větším měřítku to může být vyhýbání se uvědomnění si, že *vše co jsi doteď udělal je zbytečné, protože to právě končí*. [21]

Touhu bránit se před ztrátou prozkoumal také Daniel Kahneman [10]. Ten zjistil zajímavý poznatek a to, že lidé se obzvláště bojí o něco přijít, i když je šance velice malá. Naopak při velké šanci na ztrátu se lidé riskovat nebojí a rádi zkouší své „štěstí“. Za tímto účelem navrhnul Kahneman Fourfold Pattern 2.8. Zde je zachycené, jaké scénáře vyvolávají které emoce a touhy.

	ZISKY	ZTRÁTY
<p>VYSOKÁ PRAVDĚPODOBNOST Efekt jistoty</p>	<p>95% šance výhrát 10k \$ Strach ze zklamání AVERZE RISOVAT</p>	<p>95% šance prohrát 10k \$ Naděje vyhnout se ztrátě VYHLEDÁVÁNÍ RIZIKA</p>
<p>NÍZKÁ PRAVDĚPODOBNOST Efekt příležitosti</p>	<p>5% šance výhrát 10k \$ Naděje velkého zisku VYHLEDÁVÁNÍ RIZIKA</p>	<p>5% šance prohrát 10k \$ Strach z velké ztráty AVERZE RISOVAT</p>

Obrázek 2.8: Fourfold Pattern ukazuje, že lidé se vyhýbají i nejmenší šanci na ztrátě. Naopak při malé šanci na velkou výhru to vidí jako příležitost. Toto chování mimo jiné vysvětluje, proč se tolik daří loteriím a pojišťovacím společnostem.

Jedná se tedy o motivaci velice podbízivou, a proto není vhodné, aby na ní rozhraní stavělo. Může však být použita i k dobrým činům a výsledkům. Např. připomenutí týmu, co se může stát, pokud daný projekt nespustí včas, může motivovat k aktivnějšímu plnění úkolů. Obdobně lze ukázat možné zisky při další práci na projektu. Připomínám však, že je třeba zacházet s touto motivací velice opatrně.

Použitý jazyk

Mít dobře nastavenou komunikaci a texty aplikace dokáže změnit vnímání uživatele a dokonce jej motivovat k dalším akcím. Příkladem dobře nastavené komunikace může být Mailchimp. Jestli jste někdy posílali e-malovou kampaň, tak nejspíš znáte ten pocit úlevy,

kdy e-maily konečně pošlete. Předchází tomu totiž několik kontrol, aby se neposlala kampaň s gramatickou, nebo jinou chybou. Jelikož Mailchimp chápe svoje uživatele, podpořil tento pocit po odeslání plácnutím (viz obrázek 2.9) a tak se tento moment stává jedním z oblíbených při rozesílání kampaní.



High fives!

**Your campaign is in the
send queue and will go out shortly.**

Obrázek 2.9: Mailchimp si s uživateli plácne, poté co pošlou kampaň.

Často to jsou právě detaily, které zůstanou uživateli vryté do paměti. [1] Někdy se tyto věci zdají být až zbytečné a funkčně nic nezmění, aplikaci však chceme navrhnout pro uživatele a pocit pochopení je zde zásadní. Proto je pro menší upozornění, potvrzení a další menší interakce žádoucí udělat krok navíc, a namísto klasických strojových odpovědí nabídnout uživateli něco víc. Stejně tak důležité jsou menší animace s přímou zpětnou vazbou.

Zapojení emocí

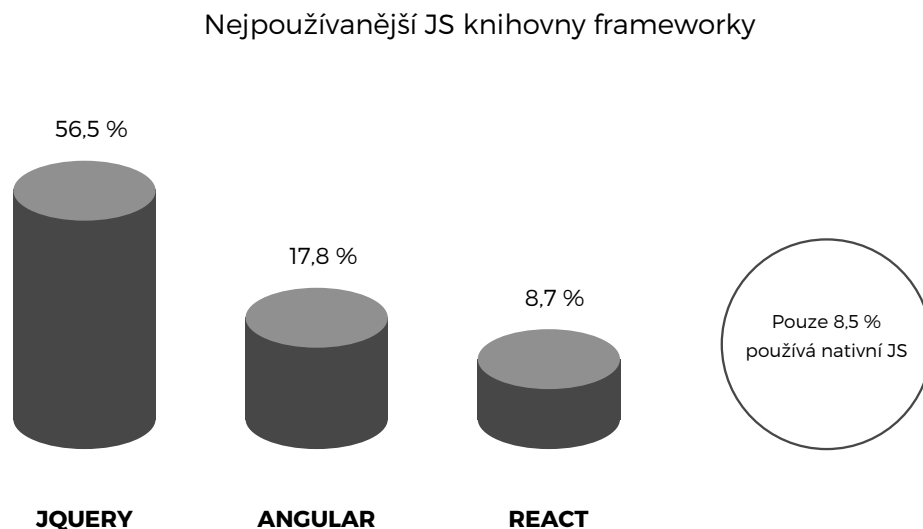
Lidé jsou emoční bytosti. Navrhování rozhraní pro emoce zapojí přirozené lidské vlastnosti a vytvoří z nich výhodu aplikace. Výsledný produkt aplikace poté uživatel nevnímá jako stroj, ale více jako osobnost, ke které má vztah. [2]

Mnoha společnostem se vyplatilo vsadit především na vizuální a psaný humor. Zkuste si například vzpomenout na nějakou z reklam, které jste v poslední době viděli. Nejspíš si

pamatujete ty vtipné a nebo ty, které ve vás vyvolaly silnou emoci, jako třeba pobouření. Stejně principy fungují také pro webové aplikace a stránky. Jako lidé si pamatujeme především momenty, kdy naše emoce fungovaly na plné obrátky. To lze využít ke zpříjemnění celého rozhraní a především k jeho zapamatování, a aby se do něj uživatelé vraceli. [4]

2.4 Výběr technologií pro realizaci prototypu

Při výběru technologií jsou první rozepsány důležité požadavky pro realizaci rozhraní. Následně jsou popsány výhody a nevýhody React a AngularJS, což jsou nejpoužívanější javascriptové knihovny a frameworky pro rok 2016. [13] Graf nejpoužívanějších javascriptových knihoven a frameworků, podle průzkumu Ashley Nolana, je na obrázku 2.10.



Obrázek 2.10: Průzkum z roku 2016 pořádaný Ashley Nolanem, kde odpovědělo 2 028 respondentů ukázal nejpoužívanější JS frameworky a knihovny. Zdroj: [16]

Jednoduchá správa a úpravy

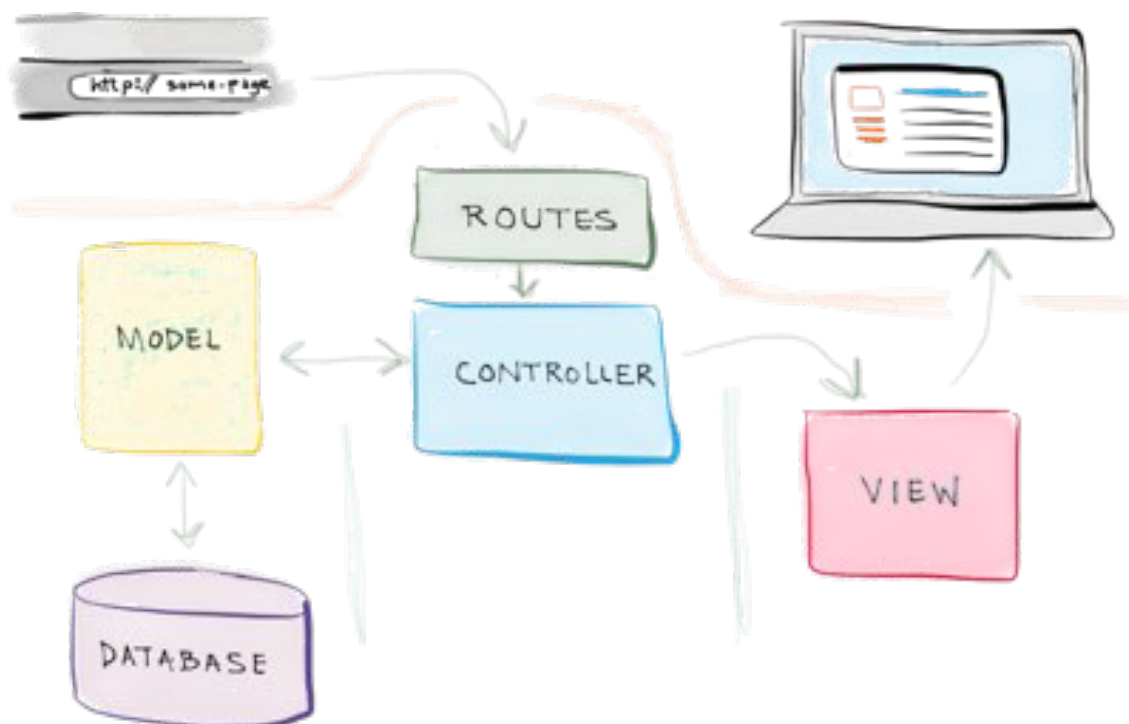
Pro jednoduchou správu jakéhokoliv kódu je důležité dodržovat styl formátování, psát komentáře a dělit kód na funkce. Mezi další dobré rady potom patří nekopírovat kód na různá místa a dělit jej na bloky s celkovým cílem napsat jej na co nejmíň řádků, ne však za cenu nepořádku. [17]

MVC

MVC je softwareová architektura, která rozděluje aplikaci na 3 vrstvy, kde M značí model (MODEL), V značí vzhled (VIEW) a C značí kontroler (CONTROLLER), což je obvykle nejdůležitější část. Díky tomuto rozdělení má každá vrstva svoje jedinečné povinnosti a o zbytek se starají vrstvy ostatní. Tedy view zajišťuje, jak bude aplikace vypadat a rozmístění jednotlivých prvků zobrazených uživateli. Model zajišťuje práci s daty. Pracuje vlastně s třídami, které mohou řídit datovou vrstvu, např. zajištění komunikace s API. A konečně controller, což je srdce celého modelu, protože dvě předchozí vrstvy propojuje. [11]

Tedy pokud uživatel pošle požadavek webovému prohlížeči, přijde první do controller. Ten to dále pošle vhodnému view a žádaná data od uživatele a controlleru jsou zadány do

vrstvy modelu, který požadovaná data obstará. Uživateli se poté zobrazí pouze view vrstva, jak lze vidět na obrázku 2.11. Hlavní výhodou je, že pokud změním jednu vrstvu MVC, ostatní vrstvy to nijak neovlivní.



Obrázek 2.11: Komunikace vrstev v MVC. [12]

Tento model zmiňuji proto, že díky rozdělení na vrstvy je jednodušší dělat v programu úpravy, a pokud se rozhodnu něco měnit, vím do které vrstvy je třeba zasáhnout a kde úpravy provést. Tuto vlastnost dále ocením především po získání zpětné vazby uživatelů, kdy se změny budou dělat především ve view vrstvě.

Velikost komunity a její aktivita

Proč se zajímat o aktivitu a velikost komunity u frameworku? Především to často značí dobrou dokumentaci a tak se snižuje čas potřebný pro jeho naučení. Je pravděpodobné, že spousta problémů s vývojem byla v rámci komunity řešena. Dobrým ukazatelem oblíbenosti frameworku bývá i počet projektů, kde je aktivně používán. Výsledky průzkumu oblíbenosti na twitteru lze vidět na obrázku 2.12

Pro zjištění velikosti komunity a aktivit je dobré položit si tyto otázky: *Kolik lidí o tom píše na webu? Kolik lidí to stále aktivně používá? Jak často developeri odpovídají na problémy? Kolik je pull requestů na gitu?*

AngularJS 2

AngularJS je open-source javascriptový webový framework. V současné době je největším přispěvatelem společnost Google, která tento framework v roce 2009 založila. Je aktivně používán na stránkách jako jsou **Forbes**, **Roblox**, **RyanAir** a dalších (celý seznam na **Lib-**

score). Jedná se o framework, který má v sobě zahrnuto množství potřebných funkcí, začít tak s AngularJS je jednodušší a potřebné funkce lze lehce najít. [9].

AngularJS silně podporuje TypeScript, což je open-source programovací jazyk. Jedná se vlastně o nadstavbu JavaScriptu, která jej rozšiřuje o statické typování a další atributy z objektového programování.

Výhodou komplexního frameworku udržovaným Googlem může být jeho budoucnost, kterou lze svěřit do rukou Googlu a dokud to tak bude, budou aplikace z AngularJS používat moderní a kvalitní metodiku. Celkově je však AngularJS komunitou považovaný jako vhodný spíše pro větší webové aplikace a stránky. [6]

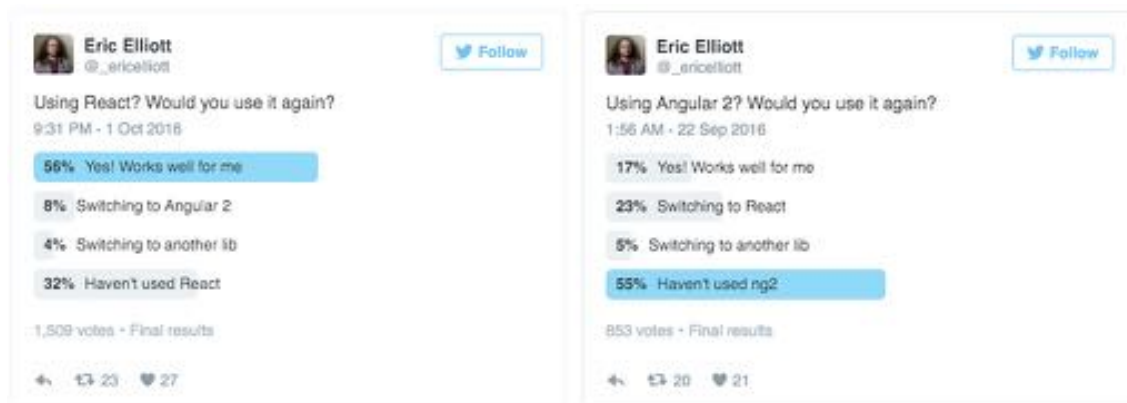
React.js

React je javascriptová knihovna pro budování uživatelských rozhraní. [3] Je aktivně používán na stránkách jako jsou Netflix, Airbnb, Walmart a dalších (celý seznam na Libscore). Jedná se tedy o knihovnu a ne framework. Nicméně díky dalším dostupným knihovnám, které k Reactu vznikly, je možné tvořit komplexní aplikace, proto jsem se rozhodl jej zde zařadit. Lze tedy říct, že frameworky nabídnou komplexní balíček služeb, zatímco s Reactem si tento balíček musíte poskládat sami.

JSX je HTML-like syntaxe Reactem používaná a je dále kompilována do JavaScriptu. Tedy značkování (eng. markup) a kód jsou skládány ve stejném souboru. Díky tomu je jednodušší se při psaní kódu odkazovat na funkce a proměnné komponentů.

React je třeba kompilovat před spuštěním, což je velmi užitečné. Znamená to, že při chybě v kódu se aplikace nepřeloží a na chybu upozorní. Kodérovi zároveň vypíše řádek chyby, čímž se urychlí proces vývoje.

JavaScript-centric design Reactu má několik výhod. Jednou z nich je, že není třeba učit se novou syntaxi, jelikož vše vychází z Javascriptu. V praxi to znamená, že se píše HTML do JavaScriptu, tedy je možné jednoduše používat funkce pro loopování a zanořování. Má to však i své nevýhody. Pro rychlejší a jednodušší syntaxi se často používá ECMAScript (ES), který se v posledních letech stále vyvíjí. Při procházení tutoriálů k Reactu tak lze najít tutoriály v ES5 i ES6 a rozdílná syntaxe může být pro začátečníka matoucí. [9].



Obrázek 2.12: Porovnání zájmu pracovat znova s React a s AngularJS. Jedná se sice pouze o data sdílená na Twitteru, přesto hlasovalo přes 2 000 uživatelů. [6]

Kapitola 3

Návrh webového rozhraní motivující uživatele

V kapitole popisuji, jak je v kontextu této práce chápán klient a designér a jak funguje proces jejich spolupráce. Dále se zabývám tím, kde v tomto procesu může hrát roli uživatelské rozhraní, a jak by mohlo celý proces usnadnit a vést k dlouhodobé spolupráci a lepším výsledkům společného projektu. Je zde definovaný hlavní cíl projektu a popsání kroky vedoucí k jeho splnění.

Dále zde popisuji, jak uživatele vtáhnout do příběhu, který ho bude provádět celým rozhraním. Zabývám se zde, jakým způsobem s uživatelem komunikovat a jak na něj působit. Na základě toho vzniká průvodce celým rozhraním.

Poté je popsáno užití motivačních metod z předešlé kapitoly a vznikají tak konkrétní prvky, které jednotlivé motivace mají podporovat. Na konci kapitoly je popsán vznik wireframu celého rozhraní, kde jsou vysvětleny a ukázány nejdůležitější prvky.

3.1 Proces spolupráce designéra s klientem a cíl projektu

Při návrhu aplikace bylo třeba prvně analyzovat proces spolupráce klienta s návrhářem a definovat si účel rozhraní. Nepředpokládám, že rozhraní plně zastoupí všechny způsoby komunikace mezi klientem a designérem. Je však cílem zapojit klienta do procesu spolupráce tak, aby se minimalizovala ostatní komunikace po síti (e-mail, messenger, apod.). Toto shromáždění on-line komunikace na jedno místo zabrání nedorozumněním, kdy uživatelé používají více kanálů a nejsou na všech stejně aktivní.

Pro průběh spolupráce na projektu jsem vycházel z osobních zkušeností v **Creepy studio**, kde pracuji přes tři roky a podobných projektů jsem řídil několik. Celý proces zde ve zkratce popíši, dle několika rozhovorů se tento proces téměř neliší od ostatních procesů v menších on-line agenturách. Popíši zde také, jak je v tomto projektu chápán klient, designér a společný projekt.

Klient

Zákazníkem je v kontextu této práce chápán především investor projektu, který si najal designéra, aby mu realizoval jeho on-line projekt. Zákazník v našem kontextu obvykle nemá dedikovaného člověka na marketing, či on-line a sám v tomto oboru nemá příliš mnoho zkušeností, pouze od přátel, případně z dřívějších projektů. Jenže on-line prostředí se za

poslední roky nespočetněkrát vyvinulo, a tak je jen logické, že najal designéra, aby mu s projektem pomohl.

Designér

Designérem je chápán freelancer, nebo menší on-line agentura, která se živí tvorbou webových prezentací, on-line kampaní nebo digitálním marketingem. Designérem tedy není vždy chápán jediný člověk, nepředpokládám však zároveň tým větší než 12 lidí. Designér obvykle spolupracuje s více klienty zároveň na různých projektech. Jeho cílem je zákazníkovi doručit kvalitní výstup v daném termínu.

Společný projekt

Společným projektem je chápána práce, kterou zadává klient designérovi s předem definovaným cílem. Získat kvalitní zadání, správně pochopit a definovat cíle přesto bývá často údělem designéra, který je potřebuje, aby mohl dodat kvalitní výstup. Během projektu spolu designér a klient spolupracují, designér má vědomosti a schopnosti realizovat projekt, zatímco zákazník zná skvěle svou firmu a způsob podnikání, které potřebuje designér pochopit.

Průběh spolupráce klienta a designéra

Po úvodním oslovení dochází k první schůzce designéra se zákazníkem. Schůzka bývá často osobní, ale také může být přes videohovor, pokud je vzdálenost mezi návrhářem a zákazníkem daleká. V obou případech na těchto schůzkách zákazník popisuje co dělá, jak to funguje a čeho chce dosáhnout. Úkolem designéra je pochopit podnikání zákazníka a zjistit hlavní cíle projektu, rozpočet a důležité termíny.

Poté je designérem vypracovaný krátký brief projektu (základní informace sepsané na papír), kde je uveden také cenový odhad a termíny plnění. Pokud vše probíhá dobře a zákazník i návrhář jsou připraveni spolupracovat, sepiše se smlouva a spolupráce je zapečetěna. Zde se také reálně začíná pracovat na projektu a klient by se měl poprvé dostat do rozhraní.

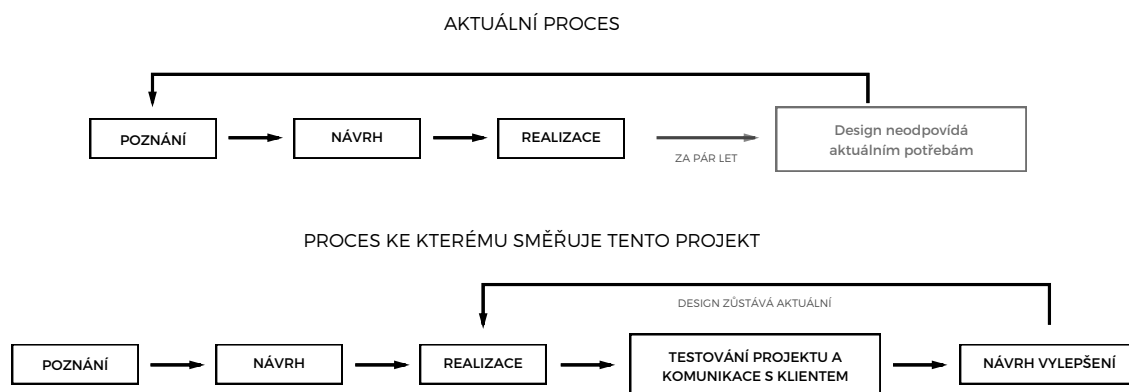
Pro jednoduchost dále představím nejčastější průběh webového projektu. Začne se **analýzou klienta**, jeho cílové skupiny a jeho konkurence. Tato fáze je kombinací zjišťování informací z dostupných zdrojů a komunikace (*často kombinování telefonátů, e-mailů a schůzek*) s klientem pro lepší pochopení koncepce a procesů jeho podnikání. Na závěr této fáze je vytvořený **wireframe** projektu, který je s klientem několikrát konzultován a postupně přepracováván. Začínají se také řešit texty, které jsou často dodávány klientem nebo externím textařem a na web dále upravovány.

Další fází je **grafika**, která obvykle přichází až po dokončení předchozí části. Na jejím počátku je vytvořený moodboard (*spousta příkladů hotových grafik vybraných designérem, které ukazují atmosféru projektu*), ten je konzultován se zákazníkem a ujasní se styl, ve kterém bude grafika tvořena. Poté jsou vytvořeny nejzásadnější stránky, které se opět konzultují, a konečným výstupem této části je poté **hotový grafický návrh webu** vycházející z wireframu.

Po dokončení fáze grafiky, přichází na řadu kód. Během této fáze se dále pracuje na textech webu, často se jedná o nejdelší část. Na rozdíl od předchozích fází zde nejsou nutné konzultace se zákazníkem ohledně stylu kódu. Pro úspěšné dokončení této fáze klient

kontroluje především funkčnost, vzhled a rychlost stránek. Po dokončení fáze kódu je web ještě testován a jsou laděny detaily, aby byl připravený na spuštění on-line.

A přichází fáze na kterou se zákazník nejvíce teší. Celý web je hotový a po několika měsících můžou konečně i ostatní vidět, na čem se pracovalo, a co stálo spoustu peněz a také času. Jenže tady by dle mě spolupráce končit **neměla**. Na web začnou chodit návštěvníci a teprve poté se ukáže, jak dobrý návrh byl a kde je možné jej vylepšit. Další spolupráce je zde pro dobré výsledky klíčová a vysvětlit to zákazníkovi bývá údělem designéra. *Zde by měl být zásadní přínos tohoto projektu, který od začátku bude směřovat k dlouhodobé spolupráci a lepším výsledkům.* Porovnání aktuálního a plánovaného procesu lze vidět na obrázku 3.1



Obrázek 3.1: Při dnes běžném modelu se projekt průběžně nevylepšuje. Časem tak neodpovídá klientovu podnikání, které se mezitím vyvíjí. Při pravidelném zlepšování projektu lze dosáhnout lepších výsledků díky sesbíraným datům a zpětné vazbě uživatelů. Designér také klienta nemusí znova poznávat, neboť chápe jeho problémy.

Hlavní cíl

Aplikace bude sloužit jako doplněk k již fungujícímu procesu, který jsem popsal výše 3.1. Jejím hlavním cílem je motivovat k dlouhodobé spolupráci zákazníka s designérem. Rozhraní jako takové pomáhá designérovi řídit svoje projekty, dále je vylepšovat a díky dlouhodobé spolupráci zajistí také stabilní příjem práce. Díky tomuto konceptu víme, že aplikace bude nabízena designérům, kteří do ní při spolupráci dále přizvou klienta. Pro splnění cíle dlouhodobé spolupráce je třeba splnit několik dílčích cílů, které dále popíši a vysvětlím, proč je považuji za důležité.

Prvním je, že si zákazník i designér zvyknou rozhraní pravidelně používat. Považuji za nutné, aby rozhraní bylo používáno od začátku spolupráce klienta a designéra. Díky tomu si klient zvykne rozhraní používat a stane se místem, kde má přehled o daném projektu. Rozhraní bude budovat prostředí, kam se klient obrací pokud potřebuje informace o projektu, a kde vypracovává svoje části zadání. Na základě vytvořeného zvyku rozhraní používat, v něm budu moci klienta postupně motivovat k další práci na projektu a postupně mu ukazovat, že spuštěním celý projekt vlastně teprve začíná.

Rozhraní bude **funkční**, to znamená, že kromě motivace k dlouhodobé spolupráci musí aplikace fungovat pro komunikaci týmu zákazník a designér. Mělo by být jasné, kdo má momentálně jaké úkoly a do kdy budou splněny, aby se celý projekt včas a kvalitně realizoval. Ideálně by mělo být možné v rozhraní obsáhnout celou on-line komunikaci. Zabraní se tak používání e-mailů, kde se ztrácí kontext celého projektu a řešených úkolů. Rozhraní si

však neklade za cíl nahradit telefonáty a osobní schůzky, během těch je komunikace často intenzivnější a umožňuje lepší pochopení a vznik nových nápadů.

Klient i designér by měli mít dobrý pocit z používání rozhraní. Tato zřejmá vlastnost je velice důležitá, neboť umožňuje jednodušeji a přirozeněji plnit dané cíle. Rozhraní nesmí uživatele mást, neboť by to odvádělo pozornost od spolupráce týmu klient a designér. Namísto spousty funkcionalit by mělo nabízet pár hlavních, které jsou stěžejní pro spolupráci. V neposlední řadě by rozhraní mělo mít svůj charakter, který na uživatele bude působit kombinací textů, barev, obrázků a animací. Uživatelé jej tak přestanou vnímat pouze jako pracovní nástroj a stane se pro ně lépe zapamatovatelné.

3.2 Příběhem do cíle

Příběhy jsou přirozenou cestou jak lidé spracovávají informace. [20] Během spolupráce na projektu se bude pracovat se spoustou informací, u kterých je důležité držet kontext, tedy vědět proč vznikají a za jakým účelem. K tomu je příběh ideální pomůcka, protože designéra i zákazníka dostane na společnou cestu za jedním cílem.

Vytyčení cílů

Každá dobrá výprava a každý epický příběh začíná společným cílem. A jak je zmíněno v sekci Epic meaning & Calling 2.3, takový cíl by měl být větší než jednotlivci na něm pracující. Mělo by jít téměř o poslání. Cíl by měl být v rozhraní často připomínán, díky němu totiž dávají všechny menší úkoly smysl, včetně těch nezáživných.

Každý projekt bude mít jiný cíl, je však důležité držet se několika pravidel při jeho definování. Zaprvé, by měla být cítit velikost výzvy. Například *mít skvělý web* je sice praktický cíl, ovšem je velice těžké si ho představit a téměř nemožné určit, kdy je splněný. Dále je důležité v cíli také oslovit klienta, použít jeho jméno nebo název jeho firmy. Použitím názvu firmy se celý tento cíl stává osobním, čímž se aktivuje motivace vlastnictví 2.3. Ukázkovým cílem, který je dost vyzývavý a osobní, zároveň může být *Pojďme Česku ukázat, kdo je to název společnosti*.

Časová osa rozhraní

Příběhy i projekty se dějí v čase a samotný proces tvorby je stejně důležitý jako finální výsledek. Pro toto rozhraní jsem tedy navrhnul časovou osu jako hlavní bod rozhraní. Každé akci a každému naplánovanému úkolu se tak musí přiřadit čas a datum. Díky tomu bude moci uživatel vidět, jaká práce ho ještě čeká, a zároveň ho uspokojí pohled na již splněné úkoly. Časová osa je zobrazena na obrázku 3.2

Píšeme si vlastní kapitoly

Mít celý projekt založený jen na jedné dlouhé timeline je však poměrně nepřehledné, podobně jako číst bakalářskou práci bez nadpisů a podnadpisů. S ohledem na průběh spolupráce klienta a designéra se celý projekt dělí na jednotlivé fáze, nebo přesněji kapitoly v příběhu.

Kapitoly jsou stejně jako úkoly zaznamenány na časové ose. Není však striktně dané, že další kapitola může začít, až ta předchozí skončí. Mohou se prolínat. Kapitoly jako takové mají vždy trochu jiné zaměření. Rozdělení není pouze na časové úseky, ale spíše na tématické bloky. Kapitola *grafika* tak může mít za cíl vytvořit grafický vzhled projektu a úkoly v ní se

budou týkat právě grafické stránky. Aby se mohla začít plnit kapitola *grafiky*, předcházely jí kapitoly *seznamování* a *wireframu*, z kterých čerpá informace.



Obrázek 3.2: Návrh časové osy vysvětluje klientovi proces práce na projektu. Uvidí zde také, kolik práce je už hotové a může sledovat, jak se blíží k dokončení.

Tato časová osa pracuje s hnacími náboji vlastnictví 2.3 a vlastní vývoj 2.3. Měla by také klientovi poskytnout dobrou představu o tom, co vše je třeba vytvořit pro dokončení projektu. Umožňuje dále plánovat projekt. Přidat další kapitolu je totiž jednoduché, a rozhraní k tomu může klienta dále motivovat, což podpoří dlouhodobou spolupráci.

Oživení ducha, průvodce aplikací

Jak jsem již zmínil dříve, aplikace vzniká primárně pro Creepy studio, kde bude testovaná a na základě toho dále upravovaná. Duch je logem a zároveň avatarem Creepy studia. Je užíváný pro veškerou on-line komunikaci, má určený svůj charakter a způsob vyjadřování. Je to trochu arogantní blázen, který si ze sebe umí udělat srandu. Pro tento projekt se duch stává také průvodcem aplikací, kterým bude po celou dobu tvorby. Jeho hlavním údělem je dávat zpětnou vazbu uživatelům a podporovat je v tvorbě. Ukázky nálad ducha jsou zobrazeny na obrázku 3.3

Pro aplikace jsem užil ducha ze dvou důvodů, prvním je branding Creepy, druhým je ozvláštňení komunikace a vložení zábavného prvku do rozhraní.



Obrázek 3.3: Duch je avatar aplikace a má svoje nálady, které mění na základě aktivity uživatelů.

Způsob komunikace

Jak celá aplikace komunikuje s uživatelem teď bude záviset na duchovi. Jak jsem již zmínil, je to trochu ulitlý blázen a idealista. Zároveň je možná také částečně parodií na designéra. Duch uživatelům tyká, chce být jejich partnerem a pomáhat jim. Příklady, jak bude duch komunikovat, lze vidět na obrázku 3.4.

3.3 Přidání motivačních prvků

V tomto bodě by klientovi měl být jasný plán projektu. Měl by vědět, co jej čeká a jaké úkoly má momentálně plnit. Chci však ještě více podpořit hladkou spolupráci na projektu tím, že rozhraní bude uživatele motivovat k plnění úkolů v daných termínech. Zde se snažím zapojit především hnací náboje vývoje 2.3 a sociálního působení 2.3.

Progress bar - touha mít sto procent

Ukazovat uživatelům, jak se posouvají společně k cíli má několik dobrých důvodů. Jednak je vidět, co vše už museli udělat, aby se dostali do současného stavu. Zadruté, jak Dr. Hugo Liu zmiňuje ve svém článku *Ukazuje se, že po dokončení komplexního úkolu mozek uvolňuje masivní množství endorfinů*. [15]. Náš mozek se tedy těší na to, až progress bar bude na sto procentech.

Touhu plnit progress bar využívám především v jednotlivých kapitolách. Zde chci uživateli ukázat, jak se mu daří, a s každým dalším splněným úkolem jej posunout blíže k naplnění baru.

Progress bar ale podporuje taky další motivaci a tou je vlastnictví 2.3. Uživatel totiž vidí, kolik toho již udělal a kolik času *zaplatil* pro dosažení cíle. Tuto motivaci podporuje především velký progress bar celého projektu umístěný v hlavičce aplikace. Ten dává motivaci plnit úkoly v jednotlivých kapitolách s vidinou splnění většího cíle.

Tým pracuje, přidej se

Rozhraní budou vždy používat minimálně dva uživatelé (klient a designér), obvykle to však na obou stranách může být více uživatelů. To je z hlediska motivace uživatelů výhoda. Jak je zmíněno v kapitole 2.3, sociální ovlivnění je mocný hnací náboj.

Rozhodl jsem se tuto motivaci podpořit. Během plnění úkolů jsou vidět i úkoly ostatních členů týmu, aby uživatel viděl, co už je hotové. Pokud uvidí ostatní hotové úkoly v i pozdějších termínech, může to vyvolat řetězovou reakci toho, že *nikdo nechce být ten, na koho se čeká*, proto bude úkoly plnit dříve.

Chtěl jsem také vytvořit pocit real-time spolupráce, kdy lidé spolupracují zároveň a navzájem se tak motivují. Proto jsem v rozhraní navrhnul malé pop-upy, které se ukáží při jakékoliv aktivitě někoho z týmu. Volil jsem spíše nenápadnou formu, protože jejich cílem není využívat úplnou pozornost uživatele, ale připomenout mu, že se na projektu pracuje.

Odměny za úsměv

Plnění úkolů by mělo také přinášet uživateli hodnotu. Dropbox poskytuje místo na uložení zdarma a LinkedIn vysvětluje, že při vyplnění profilu pravděpodobněji dostanete lepší nabídky práce. Pro tento projekt bude použita spíše technika LinkedIn, kdy je se splněním úkolů připomenut posun k cíli (viz progress bar 3.3).

Darovat klientovi slevy nebo hodnotné dary by bylo finančně náročné a nejedná se o službu, za kterou mu klient platí. Snažil jsem se tedy vymyslet odměny v digitálním světě, které jsou zdarma. Nechtěl jsem však do rozhraní přidávat vlastní měnu nebo odznaky, nejedná se totiž o rozhraní, kde by klient mohl budovat své digitální bohatství a pak jej porovnávat s ostatními, protože v rozhraní bude vždy on a designér a tyto dva týmy nechci stavět do konkurenčního vztahu.

Navrhnul jsem tedy systém odměn založený na hnacím náboji nepředvídatelnosti 2.3, který by měl podpořit nával endorfinů při dokončení úkolu. S každým dokončeným úkolem a kapitolou v projektu se uživateli ukáže na několik vteřin jásající gif. Tyto gify budou náhodně generovány a uživatel tak uvidí jako svou odměnu pokaždé jiný gif. Věřím, že dobrý výběr gifů dokáže vyvolat pozitivní emoce. Snímky z gifů jsou zobrazeny na obrázku 3.5

Také společně s využitím motivace nedočkavosti 2.3 a vlastních úspěchů 2.3 bude po dokončení úkolů zobrazen virtuální high-five, který bude vždy náhodně upraven textově

i vizuálně, tedy bude dostupná menší databáze těchto high-fives, ze které se při každém splnění úkolu bude vybírat.

3.4 Tvorba wireframů

Wireframe je vizuální reprezentace rozhraní, používá se především v ranných fázích tvorby aplikací a webových stránek. Obvykle se tvoří černobíle, aby grafická stránka neodváděla pozornost od důležitého, což je ukáзка funkcionality a uživatelské flow v rozhraní.

Nadhled rozhraní – hlavička

Hlavička aplikace běžně funguje jako rozcestník pro uživatele. Zažitým zvykem je, že zde uživatel najde nejdůležitější odkazy a možnost editovat svůj profil. Rozdělil jsem hlavičku na dvě pomyslné části. V horní části je hlavní cíl projektu, tak aby byl pro uživatele pořád viditelný a všechny jeho další akce směřovaly k tomuto cíli. Na pravé straně poté najde svůj profil. U webových aplikací je standardem umísťovat profil uživatele a jeho správu do pravého horního rohu. Ve spodní části hlavičky je časová osa projektu, která je rozdělená do jednotlivých kapitol, které jsou hlavními celky celého rozhraní. Celé rozhraní tak postupně snižuje míru abstrakce, kdy nahoře je vidět hlavní cíl, pod ním jednotlivé kapitoly, které vedou k jeho vykonání a dole samotné úkoly, které je třeba vykonat. Časová osa je tedy hlavním prvkem rozhraní, podle kterého bude uživatel navigován. V časové ose je možné taky sledovat posun v projektu. Uživatel by zde měl vidět kolik z jednotlivých kapitol je již hotových. K tomu jsou zde navrhnuté progress bary a systém notifikací, který ukazuje kolik úkolů je ještě třeba splnit pro dokončení kapitoly.

Orientace v kapitole

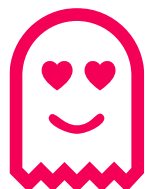
Hlavní pracovní prostor bude právě v jednotlivých kapitolách, do kterých se uživatel může dostat z časové osy v menu. Zde jsou hlavním prvkem úkoly, které je třeba vykonat pro dokončení kapitoly. Jako hlavní prvek jsem úkoly umístil na levou stranu, více se o nich rozepíši v další sekci. Na pravé straně je poté navrhnutý přehled dané kapitoly.

Přehled kapitoly slouží k motivaci uživatele pracovat usilovně na jejím plnění. Hned nahoře vidí kolik procent z kapitoly už splnil a zapojují se tak motivace progress baru a touhy dokončit úkoly (3.3). Pod přehledem je ukázán celý tým, který na projektu pracuje. Uživateli je tak připomenuto, že ostatní na něj spoléhají a umožňuje mu to kontaktovat ostatní členy týmu. Na konci je seznam již hotových úkolů, který si uživatel může postupně plnit právě jejich plněním.

Srdce aplikace – úkoly

Při způsobu spolupráce jsem se inspiroval u ostatních aplikací zaměřujících se na spolupráci v týmu. Hledal jsem něco jednoduchého, co je už dobře zažité. A to jsou úkoly. Je to vcelku jednoduchý a dobře známý koncept, neboť téměř každý si někdy udělal to-do seznam.

Při návrhu úkolů jsem se zaměřil především na jejich jednoduchost a plnění. Myslím si, že každý úkol musí mít svůj termín a zodpovědnou osobu. Pokud u úkolu není termín, pak je na jeho splnění vlastně neomezený čas a nemůžeme tak nikoho vinit, pokud jej nedokončí. Stejně jako projekty i úkoly musí mít svoje termíny. Obdobně je třeba určit, kdo je za daný úkol zodpovědný, jinak může nastat situace, kdy úkol už je sice po termínu, ale nebylo dané, kdo jej má vykonat a tak jej neudělal nikdo. Při vytváření úkolů tyto požadavky sice stojí



Wow!

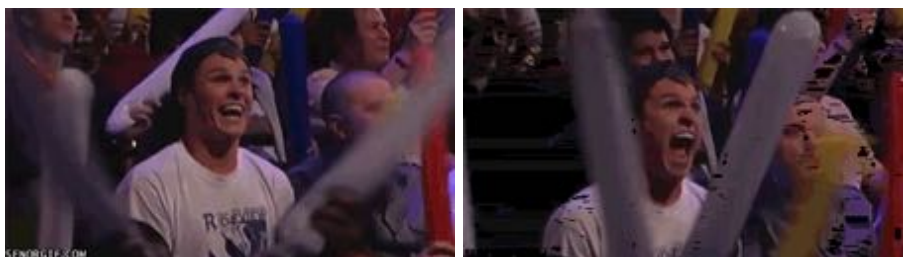
Už druhý týden zvládaš
všechno v termínech.



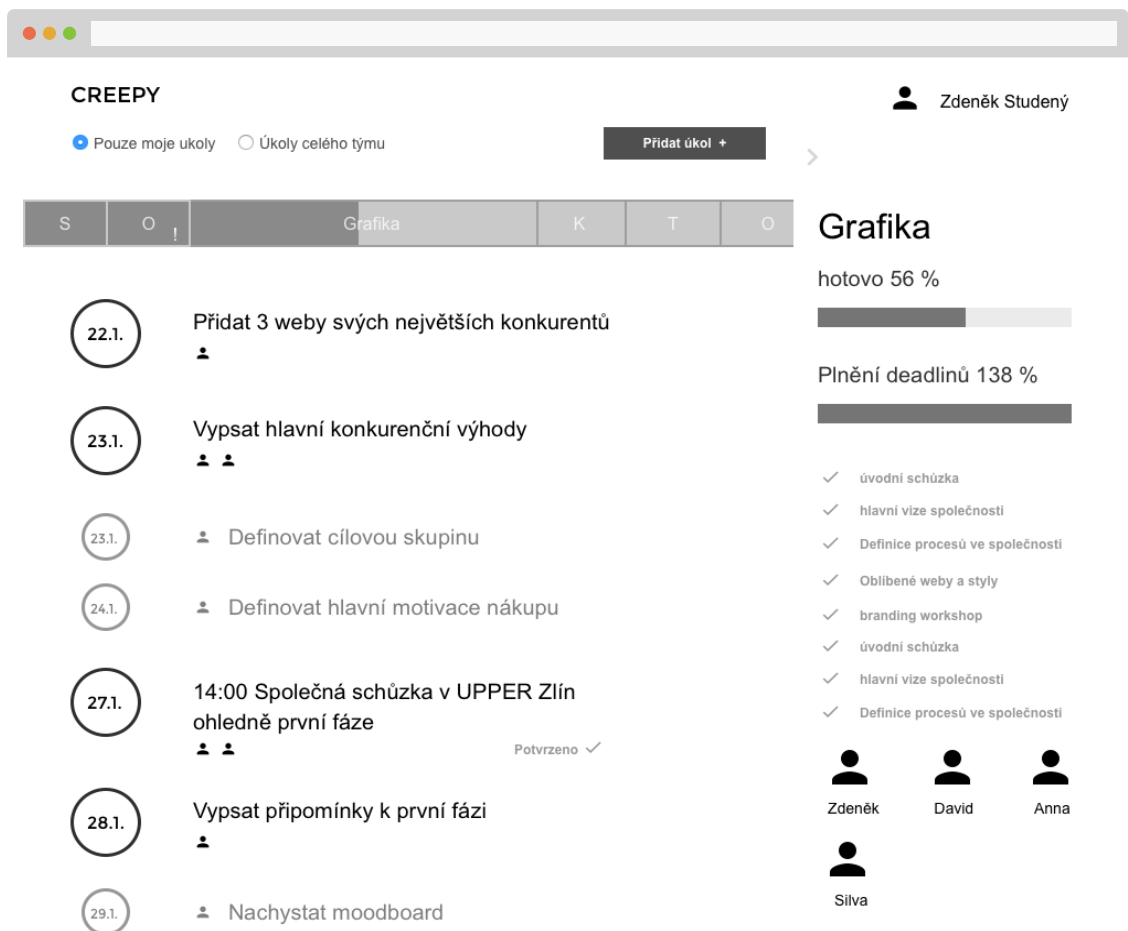
Bububu!

Týden jsi tu nebyl, bylo
mi smutno.

Obrázek 3.4: Ukázka komunikace ducha. Na levém obrázku je duch nadšený, protože uživatel zadávaný text naformátoval a tak je přehledný. Na pravém obrázku duch vyvolává pocit viny, protože uživatel projekt nekontrolovat pravidelně.



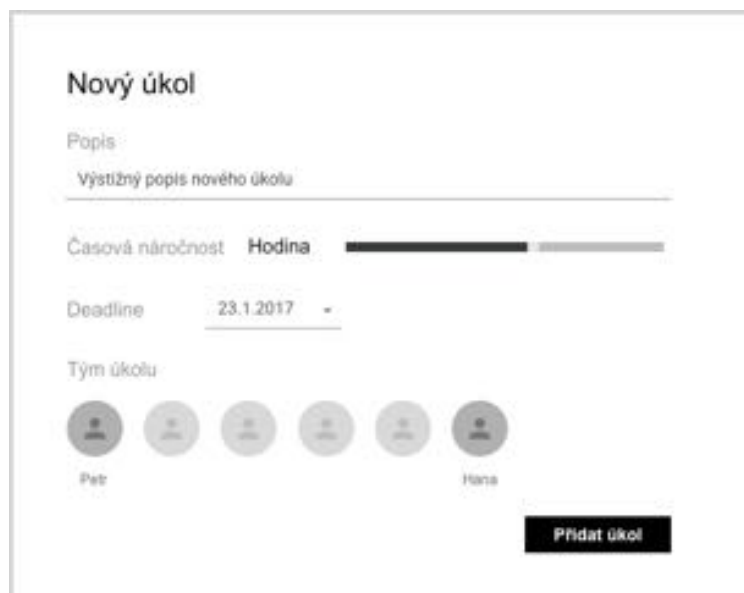
Obrázek 3.5: Dva snímky z animovaného gifu. Oproti obrázkům a textům jsou gify v pohybu. Dokáží tak dobře vyjádřit pocity a nadšení. Zdroj: [Giphy](#)



Obrázek 3.6: Návrh rozhraní, kde jsou hlavními prvky jednotlivé karty úkolů, u kterých je zásadní termín dokončení pro hladký průběh projektu.

uživatelé několik kliknutí navíc, ale při celé práci na projektu to dokáže ušetřit celkový čas. Náhled wireframu kapitoly s úkoly je ukázán na obrázku 3.6.

K jednotlivým úkolům je dále možné vést diskuzi. To umožňuje uživatelům postupně úkol vypracovávat a zároveň jej konzultovat s ostatními členy týmu, kteří zde můžou připsat svoje poznatky. Diskuze mezi uživateli jsou tak vždy tématické k danému úkolu a k dané kapitole, což by mělo usnadnit přemýšlení nad danými problémy, jelikož jsou zasazené do kontextu a uživatelům by měl být jasný celkový cíl projektu.



Obrázek 3.7: Časová náročnost úkolu pomůže uživateli odhadnout vhodný termín úkolu.

Přidávání úkolů

Srdcem aplikace je plnění úkolů, zaměřil jsem se proto na způsob jejich přidávání. Přidávání úkolu by mělo být především jednoduché, aby uživatel přemýšlel nad dobrou formulací úkolu a neztrácel čas orientací v aplikaci. Je důležité zmínit, že rychlost přidávání úkolů zde není největší prioritou, protože chci, aby uživatel při vytváření úkolu zvážil jeho zadání a usnadnil tak řešiteli jeho vykonání, protože obecná zadání nejsou vždy dobrá.

Častým problémem plánování úkolů bývají nereálné termíny, které způsobí, že se aplikace zaplní úkoly, které uživatel nemůže stihnout splnit. Proto jsem navrhnul časovou náročnost úkolů, která by měla donutit uživatele zamyslet se nad náročností úkolu a v případě jeho velké časové náročnosti na daný termín neplánovat další aktivity.

U výběru zodpovědných osob mi šlo především o možnost rychlého výběru. Oproti ostatním dostupným řešením má totiž toto rozhraní výhodu, že je určené pro menší týmy. Mohl jsem si tak dovolit při výběru ukázat avatary všech z týmu a uživatel si poté jednoduše zaklikne ty, kteří mají daný úkol plnit. Návrh přidávání úkolu je zobrazený na obrázku 3.7.

Náhled projektu

Při návrhu jsem také chtěl uživateli poskytnout kompletní náhled projektu, kdy není vidět pouze kapitola, ale uživatel uvidí všechny části projektu. Tento pohled v aplikaci má především aktivovat hnací náboje úspěchu a vývoje (2.3) v kombinaci s vlastnictvím (2.3).

Uživatel se tak může podívat, co během jednotlivých kapitol stihl udělat, a jak to pomohlo jeho projektu. Tato sekce se stává ještě zajímavější v bodě, kdy je projekt on-line a dále se na něm pracuje. Zde je poté možné klientovi ukázat, jak jednotlivé kapitoly zlepšily vývoj projektu (u webových projektů to například může být míra konverze a čas strávený na stránkách). Celý tento motivační proces poté vyzývá k napsání nové kapitoly. Návrhy pro nové kapitoly zde budou již předem připravené designérem, protože ze získaných dat již ví, kde by mohly být nedostatky. Je zde však také prostor pro rozhodnutí zákazníka. Aplikace ponechává možnost zákazníkovi vybrat si jednu z designérem navržených kapitol a nebo vytvořit svoji. Možnost návrhu další kapitoly klientem je na místě, jelikož je často s vlastními zákazníky v kontaktu a tak dostává cenou zpětnou vazbu na projekt. Náhled časové osy je zobrazený na obrázku 3.8.



Obrázek 3.8: Poté co jsou klientovi představeny výhody poslední kapitoly, je ihned vyzván k naplánování nové kapitoly.

Kapitola 4

Vývoj webové aplikace a testování

Během návrhů rozhraní a tvorby wireframů jsem je diskutoval s přáteli, bylo však třeba udělat kvalitativní testování, aby se ukázalo, jestli navržený systém opravdu funguje. Aby bylo testování relevantní, potřeboval jsem jej testovat na funkčním rozhraní.

Při vývoji rozhraní jsem se zaměřil především na prvky a funkcionality, které chci testovat. Celé rozhraní je realizované formou webové aplikace, za použití Reactu a několika dalších knihoven, které urychlily vývoj a dovolily mi soustředit se na mnou navržené prvky. Aplikaci nebylo třeba za účelem testování napojit na API, veškeré data jsou uložena na vrstvě javascriptu, kde jsou při spuštění aplikace také nahrána testovací data. Aplikace dostala svůj název **S3PS** (zkratka slovního spojení *sdílený pracovní prostor pro spolupráci*).

Vytvoření webové aplikace mi dovolilo provést testování a získat nové poznatky k rozhraní. Na základě testování jsem potom vyhodnotil jednotlivé prvky a zhodnotil jejich účinnost k definovaným cílům.

4.1 Vývoj webové aplikace S3PS

Pro tvorbu aplikace S3PS jsem si zvolil React především kvůli syntaxi **JSX**, která umožňuje psát XML tagy přímo do javascriptu, a také dalším výhodám, o kterých jsem se rozepsal v kapitole 2.4.

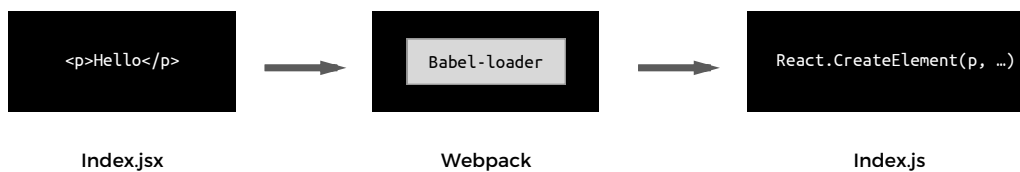
React je deklarativní knihovna pro budování uživatelských rozhraní. Komponenty Reactu zpracovávají parametry označované jako **props** a na základě dat z nich zobrazí komponent do dané podoby. Některé komponenty si samy řídí interní stav (eng. **state**), ten se v čase mění, většinou jako výsledek uživatelské interakce (například přidávání úkolu si drží interní stav zadaných dat, dokud uživatel neklikne na *uložit*, načež jsou data poslána do rodičovské komponenty). Při vývoji aplikace v Reactu je cílem mít co nejméně komponent, které mají stav, což zrychluje aplikaci a směřuje organizaci dat do návrhového vzoru singleton.

Nastavení prostředí pro vývoj

Pro využití plného potenciálu Reactu se kód píše v **ES5**, **ES6** a **JSX**. Zkompilovaný kód Reactu je však pouze čistý javascript, který je funkční pro drtivou většinu webových prohlížečů. Zároveň se React často používá společně s dalšími knihovnami, protože jak jsem zmínil dříve, nejedná se o celý framework, ale pouze o knihovnu.

Pro provázání těchto knihoven do funkčního celku jsem použil **webpack** (fungování webpacku s loadery je zobrazeno na obrázku 4.1). Ten generuje statické soubory na základě

svého nastavení. V produktové verzi aplikace pak tedy není třeba importovat několik souborů a knihoven. Není ani třeba starat se o závislosti mezi nimi. Právě závislosti se nastaví ve webpacku, který potom vygeneruje výsledný soubor, obvykle pojmenovaný `bundle.js`, který se použije v produktové verzi.



Obrázek 4.1: Webpack používá loaders pro překlad souborů do čistého javascriptu předtím, než je sváže do výstupního souboru.

Tím je vyřešené propojení knihoven a vytvoření výsledného souboru. Během vývoje by ale zabralo spoustu času po každé změně ručně překládat soubory a poté obnovovat prohlížeč, aby byly změny vidět. Naštěstí i tento problém webpack řeší a umožňuje nastavit prostředí pro vývoj, které se automaticky aktualizuje. Za tímto účelem jsem použil příkaz:

```
webpack-dev-server --progress --hot --inline
```

Pro instalaci dalších knihoven jsem používal `npm` (Node Package Manager), který mi umožnil jednoduše přidávat další knihovny a potřebné loadery. `Npm` je také vhodný nástroj pro spouštění překladů a nastavení skriptů. Součástí projektu je soubor `packages.json`, který obsahuje závislosti použitých knihoven a umožňuje definovat skripty pro překlad projektu. V konzoli potom stačí zadat jednoduše `npm start` nebo `npm build`. Zápis těchto příkazů v `packages.json` vypadá následovně:

```
"scripts": {
  "build": "webpack",
  "start": "webpack-dev-server --progress --hot --inline"
},
```

Struktura kódu

React vychází z objektové orientace javascriptu. Při práci s ním jsem se tedy snažil vytvářet soběstačné komponenty, ze kterých jsem potom skládal komplexnější komponenty a celé stránky. Jednotlivé komponenty jsou k dispozici ve složce projektu `components` a v projektu jsou potom používány tak, že se importují do hlavního souboru `app.js`.

Nahrát testovací data je možné ze souborů ve složce `test_data`. Zde jsou uložena ve formátu `json` a jsou importována do aplikace v souboru `app.js`. Pomocí rozdělení aplikace na jednotlivé komponenty a nastavení webpacku nedochází ke změně dat při úpravě komponent. Takto se změny v komponentech projeví a data přidaná za běhu aplikace zůstanou stále uložena na vrstvě javascriptu. Jediná situace, kdy je třeba aplikaci aktualizovat a nahrát data znovu ze souborů, je při editaci `app.js`.

Material-ui jako základ funkční základ aplikace

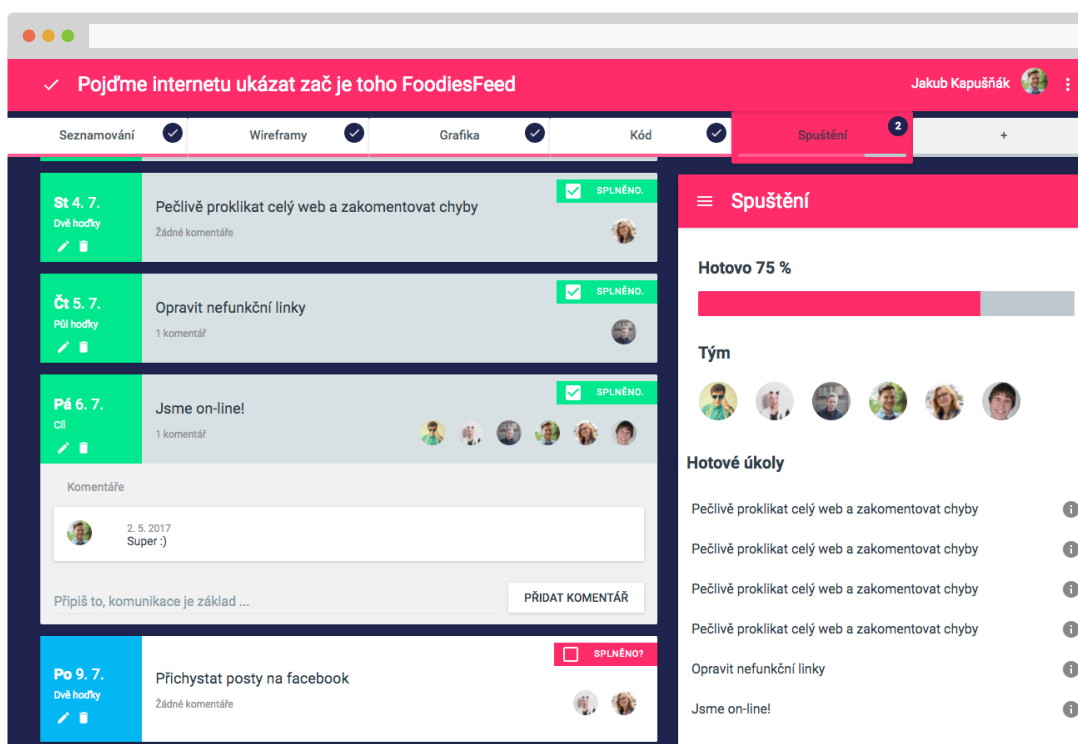
Při vývoji aplikace jsem použil knihovnu komponent pro React `Material-ui`. To mi umožnilo používat spoustu elementů s aplikovaným stylem a vlastnostmi material designu od Googlu.

Z těchto komponent jsem postavil solidní základ aplikace, stačilo už potom pouze naprogramovat vlastní komponenty, které Material-ui neposkytuje. Vytvořená aplikace s použitím material-ui je zobrazena na obrázku 4.2.

Npm umožňuje přidat do aplikace material-ui velice jednoduše, stačí v konzoli zadat příkaz `npm install material-ui` a npm se už postará o zbytek. Poté stačí v jednotlivých souborech importovat komponenty, které budou použity. Možnosti užití těchto komponent jsou zdokumentovány na stránkách [Material-ui.com](https://material-ui.com). Níže je přiložená ukázka importu komponenty `AppBar`:

```
import AppBar from 'material-ui/AppBar';
```

Pro stylování aplikace do vlastních barev jsem si definoval vlastní barevné téma, které je poté nahráno do material-ui. To umožňuje globálně spravovat veškeré barvy v aplikaci, takže další změny barev během vývoje se vždy děly rychle a pohodlně. Definovanou paletu barev lze najít v souboru `components/CreepyMuiTheme.js`.



Obrázek 4.2: Ukázka naprogramované kapitoly za použití Reactu a material-ui.

Vytvoření seznamu úkolů

Úkol je jeden z komponent, který material-ui neposkytuje. Nicméně jsem při jeho vývoji nadále vycházel ze zásad material designu, aby rozhraní aplikace zůstalo konzistentní. Jednotlivý úkol je v aplikaci definován jako bezstavový komponent pojmenovaný `TaskCard`. Z rodičovské komponenty `CreepyTaskList` dostává v `props` všechna potřebná data k zobrazení. Dále také dostává `props` (např.: `onEdit`), skrze které může volat funkce v rodičovské komponentě, to umožňuje přenos informací nahoru a zároveň dovoluje uživateli interagovat s komponentou. V `CreepyTaskList` jsou poté data úkolů mapovaná na své komponenty. Viz kód:

```

{props.tasks.map( function(task){
  return(
    <TaskCard
      date={task.due_date}
      title={task.title}
      taskMembers={task.members}
      members={props.members}
      time_level={task.time_level}
      comments={task.comments}
      key={task.id} id={task.id}
      done={task.done}
      onEdit={props.onTaskEdit}
      onDelete={props.onTaskDelete}
      onChange={props.onStatusChange}
      onCommentAdd={props.onCommentAdd}
    />
  )
})}

```

Celý úkol rozšiřuje komponentu `Card`. Levý blok úkolu skrývá časové informace o úkolu, tedy časovou náročnost úkolu a jeho termín splnění. Celý časový blok mění barvu podle náročnosti úkolu, ta je definovaná v souboru `CreepyTaskLevelStyles`. V pravé části úkolu je samotný popis úkolu společně s odpovědnými členy týmu. Ti jsou do úkolu mapováni a zobrazení jako komponenty `<Avatar>`. V této části je také přidán `<Checkbox>` pro zobrazení, zda je úkol hotový. Celou kartu lze také rozkliknout a dostat se do sekce komentářů, které jsou vedeny jako jednotlivé `Cards`. Pro stylování úkolů jsou použity in-line styly, stejně jako je tomu u `material-ui`. Karta úkolu je zobrazena na obrázku 4.3



Obrázek 4.3: U úkolu je jasně vidět, do kdy musí být splněn a kolik času si na něj vyhradit. Podle časové náročnosti se mění jeho barvy od světlých po tmavší.

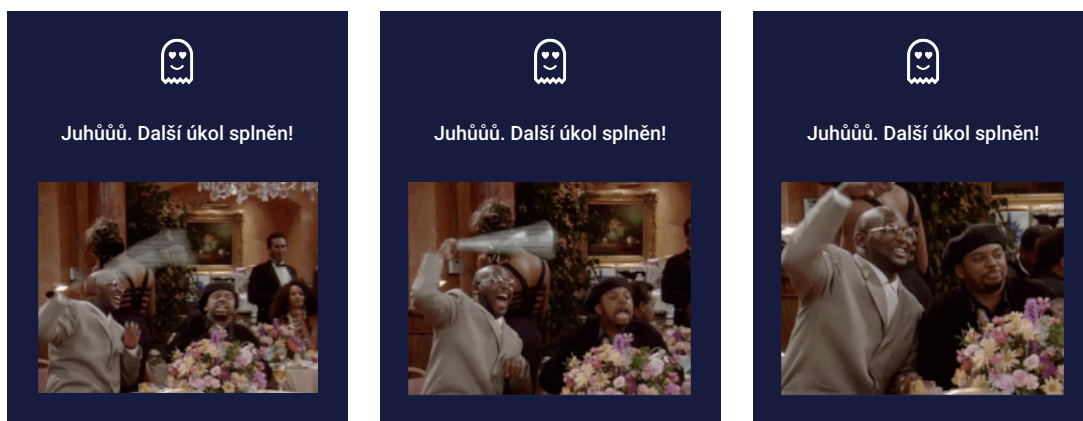
Přidání náhodných odměn

Při zpětné vazbě aplikace S3PS na akce uživatele by se uživateli kromě textu měl ukázat také tématický gif podle nálady zprávy. Nechtěl jsem vytvářet svoji databázi gifů, protože by se časem opakovaly a uživatele omrzely. Hledal jsem tedy zdroj odkud gify čerpat a

nakonec použil jeden z největších gifových portálů **Giphy**, který má otevřené API. Na portál přibývají neustále nové gify a API je generuje na základě dočasné uživatelské oblíbenosti. Díky tomu zůstávají aktuální a relativně kvalitní.

V aplikaci jsem vytvořil funkci `requestGifAndFireSnackBar(msg, gifTag)`, která bere dva parametry. První parametr obsahuje zprávu zobrazenou v pop-upu. Druhý parametr očekává tag, podle kterého bude gif vybrán (Pro oslavný gif by se volil tag *celebrate* a pro smutný *sad*).

Funkce nejprve pošle dotaz na API giphy a uloží url gifu s šířkou 200 px do stavu aplikace. Poté co je získána url gifu, se nastaví stav snackbaru na `open`, což způsobí, že ze spodní části aplikace vyjede menší infobox se zprávou a vygenerovaným gifem.



Obrázek 4.4: Při splnění úkolu se uživateli ukáže pochvala a společně s ní oslavující náhodný tématický gif. V gifu jsou obsažena krátké videa, které dokáží dobře zachytit emoce.

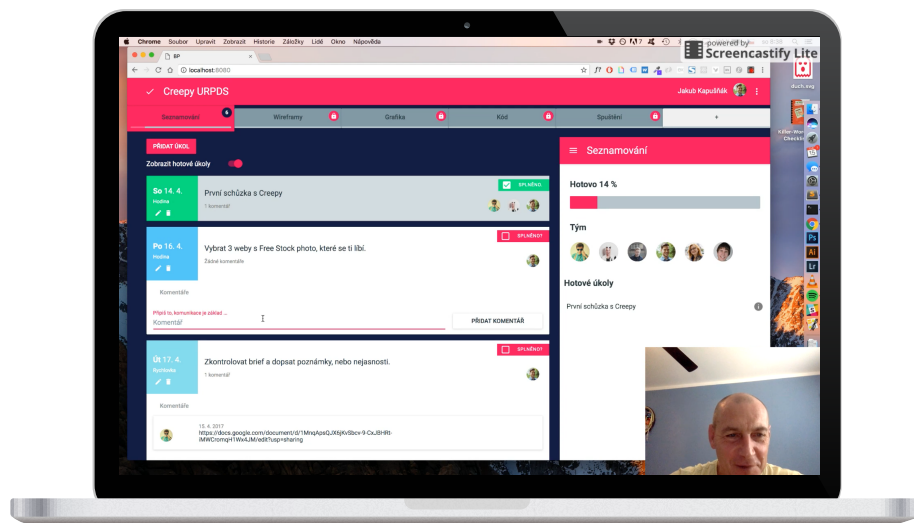
4.2 Uživatelské testování aplikace S3PS

U testování aplikace jsem se zaměřil především na to, jak uživatelé chápou celý koncept rozhraní a jestli jsou schopni v něm spolupracovat. Dále jsem hodnotil, jak navržené prvky vedou k uživatelské spokojenosti a zda motivují k dlouhodobé spolupráci na projektu. Provedl jsem dva typy testování, jedním bylo moderované hloubkové testování s následnou diskuzí po jeho skončení. Druhým typem bylo nemoderované testování, kde měli testéři vykonat sadu několika úkolů a poté vyplnit dotazník. Výsledky testování, zhodnocení aplikace a jednotlivých prvků je dále rozebráno na konci kapitoly.

Kvalitativní testování na reálných datech

Pro testování jsem se rozhodl zrekonstruovat jeden z již hotových webových projektů v **Creepy studiu**. Data použítá při testování tak reprezentují reálné situace, ke kterým dochází během spolupráce designéra a klienta. Testéři postupně naráželi na problémy, které vznikají během spolupráce na tvorbě webového projektu, a byli požádáni je řešit pomocí aplikace. S každým testerem jsme prošli postupně tři scénáře projektu, kdy každý scénář byl situovaný do jiné fáze realizace projektu. První byl začátek spolupráce s designérem, kde poprvé začal používat aplikaci S3PS. Ve druhém byl tester uprostřed realizace projektu a probíhala aktivní komunikace klienta s designérem. Ve finální třetí fázi byl projekt hotový a ladily se detaily po spuštění. To mi umožnilo simulovat různé situace a zjistit jak na

ně tester bude reagovat. Celé testování bylo nahráváno pro pozdější analýzu. Nahrávání je zobrazeno na obrázku 4.5.



Obrázek 4.5: Celé testování bylo také nahráváno pro pozdější analýzu zmatečných momentů. To mi také umožnilo sledovat reakce na zobrazení gifů a emoce při používání aplikace.

Testeři se po chvíli v aplikaci S3PS zorientovali a začali řešit úkoly. Ukázalo se, že S3PS je pro ně přehledná a pro obvyklé části projektu dostačující, chyběly jim zde však některé prvky, které by například ulehčili plánování společných schůzek. Testeři reagovali dobře na časovou osu projektu, která jim vysvětlila, jak práce na projektu probíhá, přestože neměli dřívější zkušenosti s podobným projektem. Plnění úkolů probíhalo podle očekávání, až na složitější úkoly, kdy se někteří testeři snažili zobrazit podrobný popis úkolu, který S3PS neumožňuje. Zajímavé zjištění bylo, že pouze jeden ze čtyř testovaných vnímal progress bar v sumarizaci kapitoly, zbytek testerů si jej vůbec nevšiml. Výborné reakce potom způsoboval pop-up při dokončení úkolů, kdy u prvního byli testeři lehce zmatení a při dalších zobrazení se usmívali na připravený gif.

Celkově testeři koncept aplikace S3PS pochopili, ač poukázali na několik nedokonalostí, které by před nasazením bylo nutné opravit. Většina navržených prvků také na testery fungovala, ač při testování se ukázaly způsoby, jak ještě posílit jejich motivaci. Tyto návrhy komponent rozeberu dále v této kapitole.

Nemoderované testování webové aplikace S3PS

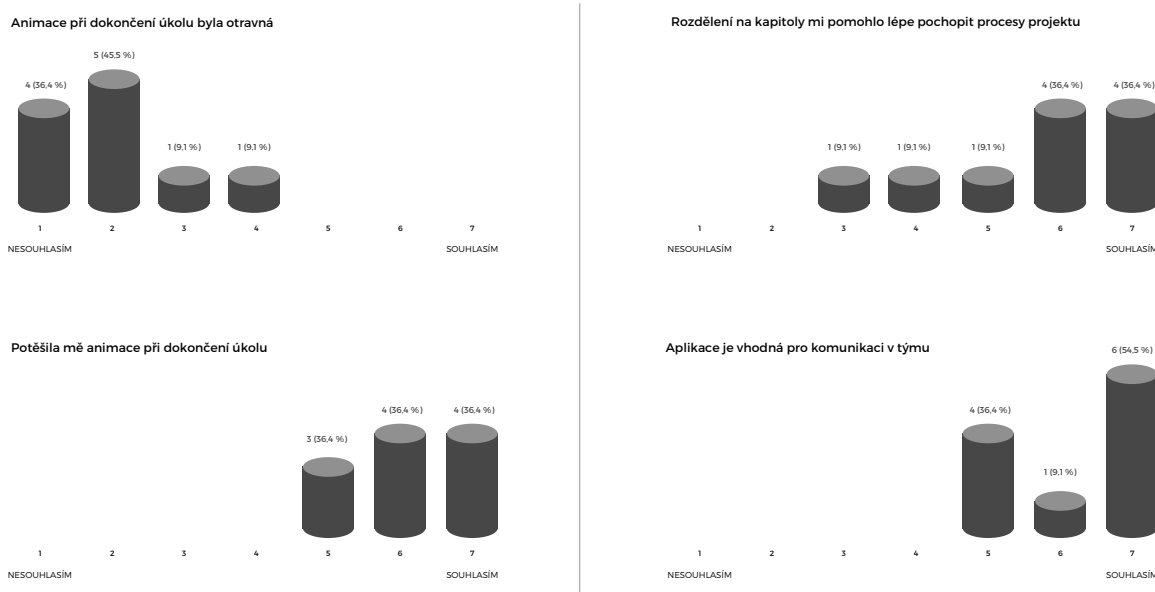
Pro otestování aplikace S3PS na větším množství testerů jsem se rozhodl provést také nemoderované testování. To sice nezjistí tolik detailů a podrobností jako kvalitativní testování, umožnilo mi ale získat větší množství dat o použitelnosti a motivaci uživatelů v navržené aplikaci.

Pro nemoderované uživatelské testování jsem nejprve připravil scénář a úkoly, které jsem následně poslal jedenácti testerům. Ti se tedy nejprve seznámili se situací, kdy jako klient poptávají nové webové stránky a za tímto účelem se rozhodli oslovit a spolupracovat s malým webovým studiem. Úkoly testery vedly k problémům a situacím, které během této spolupráce nastávají, a nutily testery k jejich řešení používat aplikaci S3PS. Protože práce na podobných projektech je časově náročná a často trvá několik měsíců, přidal jsem

zde i několik úkolů navrhnutých pouze k otestování plánování v S3PS. Na konci testování byl potom všem testovaným poskytnutý formulář s několika tvrzeními ohledně aplikace a likertovou škálou pro vyjádření souhlasu či nesouhlasu.

Výsledky testování kopírovaly kvalitativní testování, kdy se devět z jedenácti testerů v S3PS hned orientovalo a časem se tato orientace zlepšovala. Především dobrá se aplikace ukázala ve vysvětlení designového procesu tvorby webové stránky, který byl po použití aplikace pochopený všemi testovanými. Také se potvrdila špatná aplikace progress baru v jednotlivé kapitole, kterou téměř polovina uživatelů nesledovala.

Testování také vyvrátilo obavy ohledně otravnosti animovaných gifů při splnění úkolů. Žádný z testovaných je neoznačil za otravné a při otázce zda testery animace potěšily odpovíděli všichni testeři pozitivně. Osvědčila se také výzva plánovat další vylepšení projektu v časové ose na kterou klikli všichni testeři, ač záměrně nebyla zahrnuta v žádném úkolu. Výsledky jsou zobrazeny na obrázku 4.6.



Obrázek 4.6: Výsledky z nedomerovaného testování s použitím likertovi škály ukázaly, že uživatele těší gif při dokončení úkolu a vyvrátil obavy z jeho otravnosti. Testerům také pomohl pochopit proces práce na projektu a v závěru aplikaci S3PS hodnotili jako vhodnou pro komunikaci v týmu.

4.3 Rozbor prvků na základě testování

V této sekci popisují, jak jednotlivě navrhnuté prvky fungovaly na testery, a jak byly pochopitelné a použitelné. Zamýšlím se zde také nad tím, proč některé prvky na testery nepůsobily a některé splnily svůj účel. Dále navazuji poznatky a návrhy, jak by bylo možné prvky vylepšit po funkční i motivační stránce. Konkrétní návrhy na nové prvky jsou poté podrobněji popsány v další sekci.

Časová osa projektu

Časová osa (viz obrázek 4.7) se osvědčila jako dobrá pomůcka, která ukáže uživateli jak probíhá proces tvorby projektů v on-line prostředí. Díky tomu si uživatelé dokáží představit, co vše je třeba připravit a reagují na potřeby příslušných kapitol v kontextu. Jednomu testerovi však nebylo jasné, že podle časové osy by se měl pohybovat v aplikaci, což mohlo být nedostatečným pochopením toho, jak se na projektu bude pracovat. Věřím, že proces tvorby projektu by se měl vysvětlit už na společné schůzce s designérem. Není proto třeba do aplikace přidávat úvodní tutoriál pro vysvětlení časové osy. Na základě tohoto poznatku jsem se však rozhodl zvýraznit efekt při najetí myši na prvek, aby bylo uživateli jasné, že se jedná o klikatelný prvek.

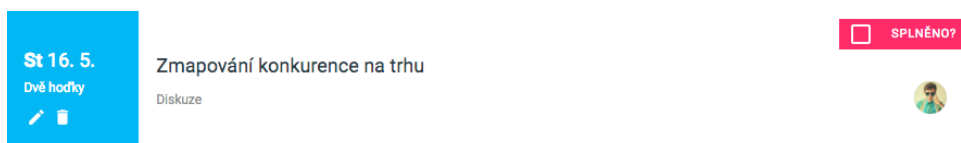


Obrázek 4.7: Časová osa pomohla testerům orientovat se v procesu práce na projektu.

Úkoly

Práce s úkoly na testery působila přirozeně. Testeři se také rychle orientovali v seřazení úkolů dle datumu a postupovali podle výrazných termínů u jednotlivých úkolů. Když jsem jako experiment nechal úkoly časově neuspořádané, způsobilo to, že testeři museli projíždět celou kapitolu, aby se ujistili, že nějaký úkol nevynechali. U přidávání komentářů testerům chyběla funkcionality automatického vytváření linků z adres stránek, to je funkce důležitá v on-line prostředí, nebylo to však zásadní pro cíle tohoto rozhraní. Časová náročnost úkolů byla vnímána jen některými testery, což dává prostor k úpravám jejího zobrazení. Osoby přiřazené k úkolu byly pochopitelné všem testerům, někteří by však uvítali avatary o něco větší. Také poukázali na možnost filtrovat úkoly pouze na ty pro sebe. Ukázka úkolu je na obrázku 4.8.

Během testování se také ukázal problém, kdy uživatel nemá zobrazené hotové úkoly a některý úkol splní. V takovém případě mu totiž úkol instantně zmizí. Docházelo poté k nejasnostem, kdy si tester nebyl jistý, jestli označil správný úkol. Pro řešení tohoto problému by bylo dobré přidat animaci postupného mizení úkolu a ve snackbaru pochvalujícím za splněný úkol zmínit také, který úkol uživatel splnil.

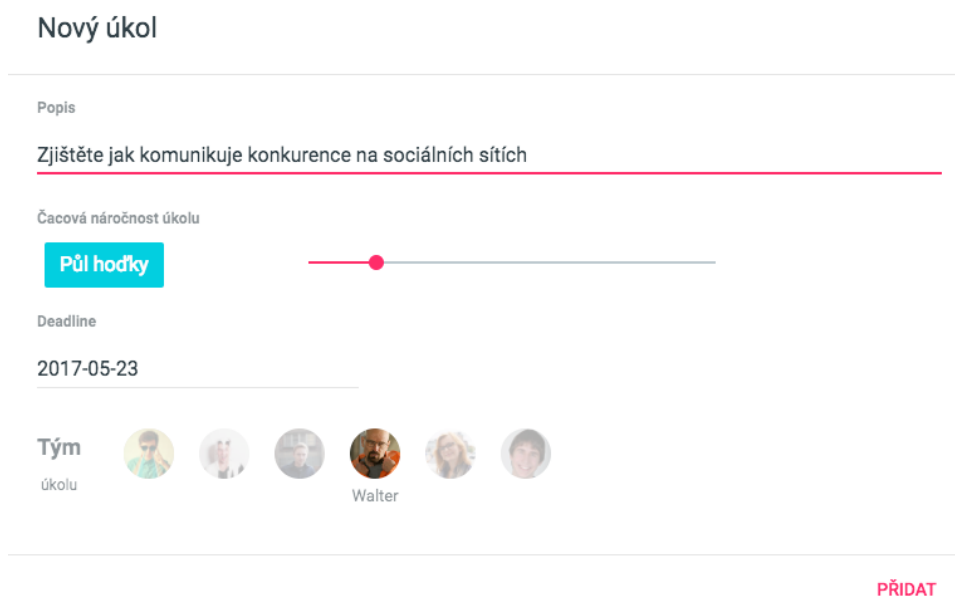


Obrázek 4.8: Karta úkolu se celkově ukázala jako použitelná. Vyznačení barvami časové náročnosti však bylo zbytečně výrazné a mnoho uživatelů nezajímalo, naopak zde chybělo zvýraznění úkolů po termínu.

Přidávání úkolů

Testoval jsem především rychlou orientaci v přidávání úkolu a způsob plánování termínů. Při zadávání časové náročnosti úkolu někteří testeři nejprve zkusili klikat na časový odhad

místo slideru. Zde by bylo dobré při kliku změnit hodnotu slideru, aby kliknutí mělo odezvu, a zároveň uživateli ukázalo, že může časový odhad měnit právě pomocí slideru. V diskusi po testování zmínili tři ze čtyř testerů, že se po zadání časové náročnosti více soustředili na výběr termínu, což je žádané chování. Výběr osob pro daný úkol probíhal rychle a intuitivně. Dva testeři však zmínili, že by při zadávání rádi viděli, kdo má jakou roli v projektu. Modální okno přidávání úkolů je zobrazeno na obrázku 4.9



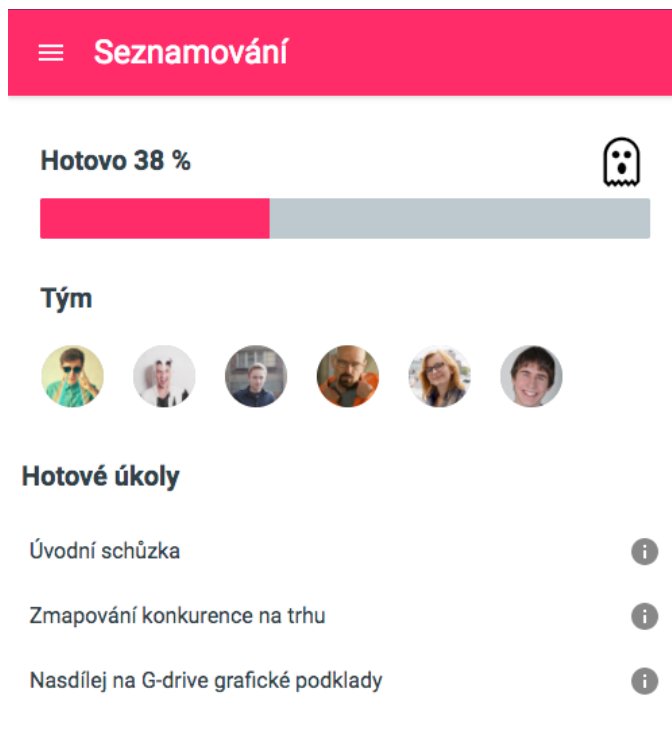
Obrázek 4.9: Jedinou komplikací při zadávání úkolu byla časová náročnost, uživatelé nejprve klikali na časový box a až poté zkusili hýbat se sliderem. Řešením by mohlo být posunout slider o jednu úroveň s každým klikem na časový box.

Sumarizace kapitoly

Sumarizace kapitoly je umístěná na pravém boku aplikace a obsahuje informace o průběhu kapitoly, kontakty na tým a shrnutí již hotových úkolů. Jejím cílem je tedy především motivovat uživatele k naplnění kapitoly, umožnit přímo kontaktovat jednotlivé členy týmu a ukázat již hotovou práci, která má motivovat na projektu dál pracovat.

Progress bar umístěný na začátku sumarizace se ukázal jako téměř neviditelný, všiml si jej pouze jeden ze čtyř testovaných. Nedá se tedy říct, že by uživatele nějak motivoval k plnění úkolů. Pro zlepšení jeho výkonnosti by jej bylo třeba vizuálně upravit a zaměřit se na ukázání posunu s každým splněným úkolem. Momentálně je totiž pohyb v progress baru příliš nevýrazný (testovaný vzhled sumarizace kapitoly a progress baru je zobrazený na obrázku 4.10). Avataři týmu, kteří po kliknutí ukáží informace jednotlivých členů, se odvěděli dobře. Testerům bylo jasné kam se orientovat, když chtěli někoho kontaktovat a ihned našli příslušné informace. Přestože v informacích je i role jednotlivých členů, někteří testeři by uvítali i více informací, například, co mají na starost, a jaké mají zkušenosti. Sledování hotových úkolů se také ukázalo jako poměrně nevýrazné a nevnímané testery. Možné řešení by bylo nahrazení jiným komponentem nebo přestylování hotových úkolů a zapojení červené barvy u úkolů splněných až po termínu, což by mohlo uživatele motivovat

k plnění tohoto sloupce pouze zelenými úkoly, které splnili včas. Během testování se také ukázalo, že při užívání na středních a větších obrazovkách je nežádoucí, aby byla sumarizace zajištěná. Jeden z testerů si ji schoval a poté nemohl najít kontakty na členy týmu. Možnost ukázání týmu by se také měla ukázat v menu. To bude stěžejní zejména pro mobilní zařízení.



Obrázek 4.10: Nejhorším prvkem S3PS se stala sumarizace kapitoly. Testery zde zajímali pouze kontakt na členy týmu. Sledování pokroku a hotových úkolů naprosto ignorovali.

Výzva k pokračování další kapitolou

Výzva k pokračování další kapitolou je umístěna v časové ose úplně vpravo a slouží k plánování dalších vylepšení projektu a tedy k budování dlouhodobé spolupráce. V jednom z prvních prototypů jsem pro možnost přidání další kapitoly používal jednoduché plus (+), což testery nijak neoslovilo. Dalo by se říct, že jej ignorovali. Pro další testování jsem tlačítko v časové ose zvýraznil a místo jednoduchého plus použil nápis **vylepšení +**.

Během dalších testování se ukázalo, že to mělo obrovský dopad. Nově na tlačítko klikli všichni testovaní a tím se dostali na možnost vybrat si, jak s projektem dále pokračovat. V nabídce, jak projekt dále vylepšit, je důležité, jak bude formulovat možné vylepšení každý jednotlivý designér používající aplikaci. Výzva se tedy ukázala dost zajímavá na to, aby si získala pozornost uživatelů, čímž hodnotím její účel za splněný.

Snackbar plný odměn

Hlavním účelem snackbaru 4.4 bylo odměnit uživatele za plnění projektu. K tomu sloužily tématicky generované gify z **Giphy**. První reakcí testerů na zobrazený gif bylo překvapení, někteří dokonce přestali pracovat dokud nezmizel. Při dalších zobrazeních se již uživatelé

nenechali překvapit a v téměř polovině případů se nad gifem lehce pousmáli. Při diskuzích po testování jeden z testerů zmínil, že to bylo příjemné odreagování a aplikace tak působila více přátelsky. Tento prvek na uživatele působil tak, jak bylo plánováno.

Shrnutí testování

Testování ukázalo v aplikaci S3PS několik nedostatků, které bude třeba před použitím opravit. Menší úpravy je třeba udělat v zobrazení úkolů, kde se barevnost dle časové náročnosti ukázala až moc výrazná (a přitom si testeři často ani barvy nespojili s náročností). Při větším objemu úkolů se ukázalo potřebné také filtrování dle autora úkolu. Jednoznačně nejhorší se ukázal progress bar v sumarizaci kapitoly, kterého si většina testerů ani nevšimla. Zde bude třeba zvážit vizuální změnu a komunikovat pokrok v kapitole také jiným způsobem, kde si jej uživatel všimne a bude si ho vědomý.

Skvěle při testování fungovalo rozdělení projektu do časové osy a všichni testeři pochopili, jak tvorba projektu probíhá a na které aspekty se v jednotlivých kapitolách zaměřit. Společně s časovou osou se osvědčila také výzva pokračovat v projektu, která zaujala všechny testery.

Obdobně dobře fungoval snackbar s pochvalou a náhodným pracovním gifem za splnění úkolu. I přes jistou obavu, kterou jsem měl vůči gifům se ukázalo, že to testery baví, a žádný z nich je neoznačil za otravné.

4.4 Návrhy nových prvků

Při testování S3PS jsem zjistil, které komponenty na testery nefungovaly tak, jak byly navrhnuty. V této sekci se zabývám změnou nebo nahrazením těchto komponent za účelem zlepšení aplikace a podpoření motivace klienta dlouhodobě s designérem spolupracovat. Současně se zabývám zlepšením použitelnosti aplikace v částech, kde vznikaly nejasnosti. Výsledkem je tak rozšíření a vylepšení původního návrhu o sadu dalších komponent, které vycházejí ze zkušenosti předešlého uživatelského testování.

Rozpracování výzvy k pokračování další kapitolou

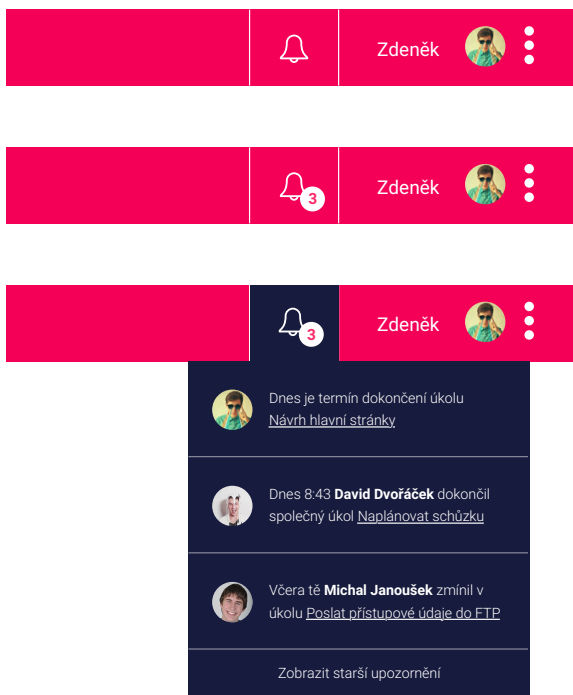
Výzva k pokračování další částí projektu se ukázala jako velmi zajímavá pro testery. Po pozdější analýze jsem zjistil, že sice klientovi dokáže představit možnosti jak pokračovat v projektu, ale nemutí jej toto rozhodnutí učinit v brzké době. Což je v praxi problém, protože designér si potřebuje naplánovat svůj čas u jednotlivých klientů a pokud by se klient na některou nabídku rozhodl zareagovat až za delší dobu, nemusel by designér už mít prostor ji splnit.

Proto jsem se rozhodl k nabídkám přidat časové vymezení od kdy jsou platné, a tím motivovat klienta k rozhodnutí, jak na projektu dál pracovat. A jelikož jsou nabídky časově omezené, je důležité na ně klienta upozornit tak, aby je nepřehlédnul. To se bude dít dvěma způsoby. Zaprvé notifikací, že je k dispozici nové vylepšení projektu a upozorněním, že se blíží expiraci. Zadruhé čítadlem v samotném vylepšení, které ukazuje kolik je právě k dispozici vylepšení. Věřím, že to bude motivovat klienta rozhodnout se, jak chce na projektu dále pracovat, a umožní také designérovi lépe naplánovat svůj čas.

Přidání upozornění

Při testování a v následných diskuzích s testery se ukázalo, že v aplikaci chybí způsob, jak uživatelům připomenout důležité události. Tento problém jsem se rozhodl řešit přidáním notifikací v hlavičce. Ty můžete vidět na obrázku 4.11 a uživatele upozorní na události, které se jej týkají.

Uživatel tak dostane upozornění vždy, když nastane změna stavu jeho úkolu a nebo úkol někdo okomentuje. Stejně tak dostane upozornění pokud se bude blížit termín některého z jeho úkolů nebo přibyla nová možnost jak pokračovat v projektu, či některá z možností brzy expiruje.

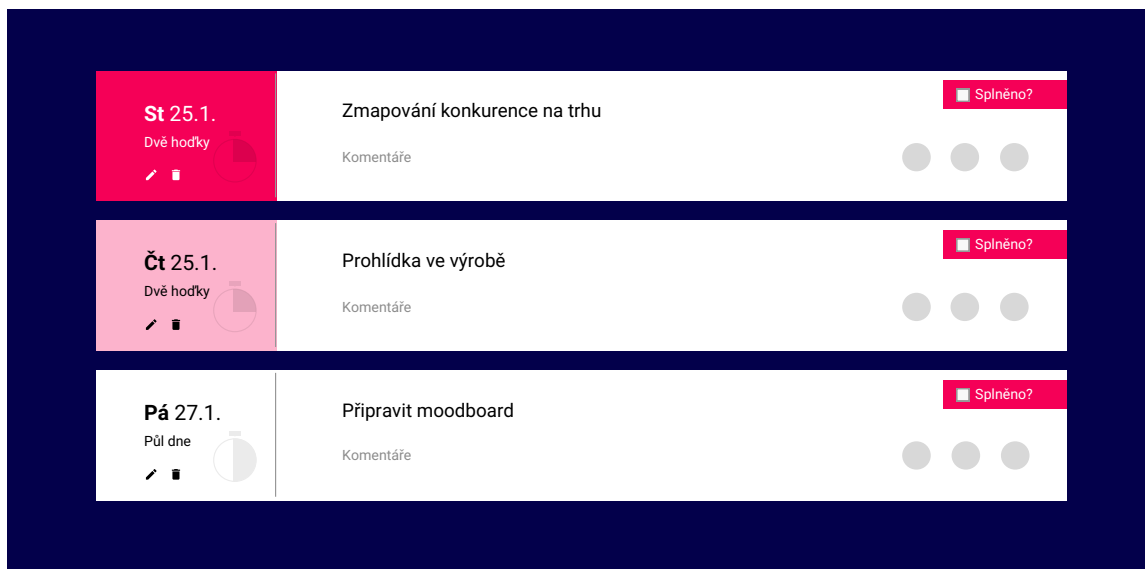


Obrázek 4.11: Upozornění informuje uživatele o událostech s ním spojených od poslední návštěvy. Připomenou mu také, aby nepromeškal blížící se termíny.

Úpravy v zobrazení úkolů

Při testování úkolů se projevila potřeba zvýraznit, kdy se daný úkol blíží svému termínu a nebo je už po něm. Vzhledem ke spolupráci je tato informace zásadní a nedodržení termínu tak dokáže zpomalit vývoj projektu. Chtěl jsem jí tedy dát velkou vizuální důležitost. Avšak přidáním dalšího výrazného prvku do současného vzhledu úkolu, by se v něm nacházelo už příliš mnoho výrazných prvků, což by ve výsledku ubíralo důležitosti všech.

Zobrazení úkolů jsem tedy redesignoval. Časovou náročnost jednotlivých úkolů nově nezvýrazňuji barvy, ale ikona stopek, která se plní dle časové náročnosti daného úkolu. Celé zobrazení úkolů je tak mnohem konzistentnější, protože už nezáří několika barvami. Pro zvýraznění blížících se a prošlých termínů u jednotlivých úkolů jsem nově použil zvýraznění barvou, kdy se zbarví celá levá část karty úkolu. Úkoly, které jsou po termínu tak nově přímo bijí do očí. Návrh nového zobrazení úkolů je na obrázku 4.12.



Obrázek 4.12: V novém návrhu karty úkolu jsem upozadil časovou náročnost úkolu a zaměřil se na zvýraznění blížících se a zaměškaných termínů. Obrázek od shora ukazuje úkol se zameškaným termínem, s termínem dnes a úkol který má termín až příští den, nebo později.

Úprava pokroku v kapitole

Testování odhalilo, že většina testerů progress bar v sumarizaci kapitoly nijak nevníkala a aktivně nesledovala svůj posun k cíli. Vytvořil jsem tedy nový návrh sumarizace kapitoly [4.13](#), tak aby nesla větší informační hodnotu a uživatelé se jí zvykli sledovat.

Důvodem, proč uživatelé nesledovali progress bar bylo, že nebyl dost výrazný. Nicméně se od něj toto po většinu práce v aplikaci neočekává. Progress bar by se měl zvýraznit v bodě, kdy je splněný nějaký úkol, a právě tehdy uživatel může sledovat posun procent a naplnění baru. Součástí progress baru je taky duch, který má výraz podle toho, jak se bar plní (při nula procentech je smutný a při sto procentech veselý). Výraz ducha se mění vždy po dvaceti procentech a to je bod, kdy chci, aby uživatel sledoval, jak se progress bar plní. Nově jsem navrhnul, aby se při změně výrazu ducha celé rozhraní ztmavilo a zůstal svítit jen progress bar, který se naplní a duch se rozradostní. Tato animace zabere něco málo přes vteřinu, a ač při každém posunu baru by mohla být otravná, věřím, že čtyřikrát za kapitolu poslouží pouze k větší motivaci uživatele.

Změna plánování schůzek

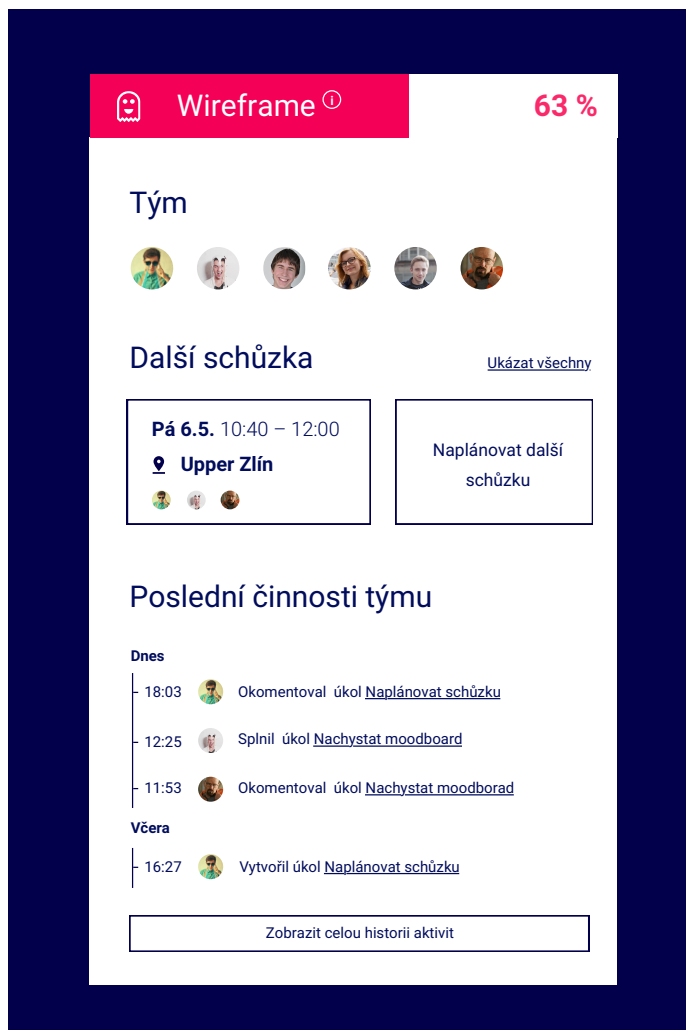
Původním nápadem bylo, že celé rozhraní bude fungovat pouze na bázi karet úkolů. Svým způsobem je totiž schůzka klienta s designérem také úkol, který musí oba splnit. Během kvalitativního testování jsem však vysledoval, jak různě uživatelé přistupovali k plánování schůzek a formou úkolu to prvně nezkusil ani jeden z testovaných.

Rozhodl jsem se schůzky od úkolů oddělit a vést je zvlášť. S redesignem sumarizace kapitoly jsem schůzky zařadil právě sem a nechal jsem odkaz také v menu, kde je několik testerů hledalo. Výsledný návrh je ukázaný na obrázku [4.13](#) a schůzky jsou nově naprosto oddělené. Četnost schůzek během projektu obvykle bývá jednou za týden, uživatelům tedy ukazují pouze příští naplánovanou schůzku, což by mělo stačit pro naplánování aktuálního týdne. K dalším schůzkám se uživatel případně může prokliknout.

Přidání sledování aktivit týmu

Posledním prvkem v sumarizaci byly již hotové úkoly. Ty měly uživatelům ukázat, co vše již dokázali a tým je motivovat k další práci. Stalo se to však spíše hluchým místem S3PS.

Nově jsem navrhnul v tomto místě zapojit motivaci spojenou se sociálním aspektem z kapitoly 3.3. Na obrázku 4.13 je navrhnutá časová osa, na které jsou promítnuty poslední činnosti celého týmu. Uživatel se tak nově může podívat, co vše se v projektu udělalo od poslední návštěvy, a může sledovat aktivitu ostatních. Touto úpravou dostala tato část aplikace kromě motivace také přidanou informační hodnotu, kdy je jednoduché sledovat veškeré události v projektu.



Obrázek 4.13: Nový návrh sumarizace kapitoly nese informace o společných schůzkách a drží uživatele v obraze záznamem posledních aktivit všech členů týmu.

Kapitola 5

Závěr

Cílem práce bylo navrhnout a otestovat uživatelské rozhraní pro spolupráci klienta s designérem, které povede k stálému vylepšování projektu. Dlouhodobá práce na projektu zajistí lepší výsledky pro klienta a stálý přísun práce pro designéra.

Pro naplnění tohoto cíle se práce nejprve zabírá návrhem uživatelsky příjemného rozhraní a dále popisuje, jak uživatele motivovat k akci za použití prvků gamifikace. Pomocí těchto znalostí poté vznikl návrh rozhraní, který si klade za cíl vytvořit u klienta návyk pracovat v rozhraní na denní bázi a následně ho vyzývá k plánování další práce na projektu.

Za účelem otestování navrhnutého rozhraní vznikl prototyp webové aplikace s použitím knihovny React, který umožnil provést sadu testování dat získaných z reálných projektů. Na základě těchto testování je popsáno a diskutováno, jak na uživatele navrhnuté prvky fungovaly.

Vzniklo tedy uživatelské rozhraní, které se za pomoci časové osy osvědčilo ve vysvětlení procesu práce na projektu. Návrh úkolů poskytuje jednoduchou formu spolupráce podobnou konceptu to-do. Zařazení výzvy plánovat další kapitoly projektu do časové osy si také získalo pozornost uživatelů, které tak přimělo přemýšlet nad dalším vývojem projektu. Celý tento způsob spolupráce doplňují uživatelsky dobře vnímané odměny za práci na projektu ve formě komentáře a náhodného tématického gifu.

Pro další vývoj vzniklo několik návrhů nových komponent, které vycházejí ze zkušeností z testování a řeší nejasné a nemotivující části aplikace. Do budoucna bude třeba tyto komponenty začlenit do aplikace. Následně aplikaci napojit na REST API a začít používat na reálných projektech, kde budu moci posbírat důležitou zpětnou vazbu a následně S3PS dál vylepšovat.

Literatura

- [1] Babich, N.: Microinteractions: The Secret of Great App Design. [Online; navštíveno 24.1.2017].
URL <https://uxplanet.org/microinteractions-the-secret-to-great-app-design-4cfe70fbaccf#4zgcpl7u2>
- [2] Cipan, V.: Designing for emotions – UX design principles. [Online; navštíveno 24.4.2017].
URL <http://www.uxpassion.com/blog/designing-for-emotions-ux-design-principles/>
- [3] documentation, R.: React. [Online; navštíveno 26.1.2017].
URL <https://facebook.github.io/react>
- [4] Dragilev, D.: Building Interfaces: How Emotion Rules Decisions. [Online; navštíveno 24.4.2017].
URL <http://zurb.com/article/827/building-interfaces-how-emotion-rules-dec>
- [5] Dubberly, H.: Alan Cooper and the Goal Directed Design Process. [Online; navštíveno 1.5.2017].
URL <http://www.dubberly.com/articles/alan-cooper-and-the-goal-directed-design-process.html>
- [6] Elliott, E.: Angular 2 vs React: The Ultimate Dance Off. [Online; navštíveno 26.1.2017].
URL <https://medium.com/javascript-scene/angular-2-vs-react-the-ultimate-dance-off-60e7dfbc379c#eb7fhfsec>
- [7] Elmansy, R.: Characteristics of Human Centered Design. [Online; navštíveno 24.4.2017].
URL <http://www.designorate.com/characteristics-of-human-centered-design/>
- [8] Ewer, T.: 10 Rules of Good UI Design to Follow On Every Web Design Project. [Online; navštíveno 16.4.2017].
URL <https://www.elegantthemes.com/blog/resources/10-rules-of-good-ui-design-to-follow-on-every-web-design-project>
- [9] House, C.: Angular 2 versus React: There Will Be Blood. [Online; navštíveno 26.1.2017].
URL <https://medium.freecodecamp.com/angular-2-versus-react-there-will-be-blood-66595faafd51#eaph517ee>

- [10] Kahneman, D.: *Myšlení rychlé a pomalé*. Jan Melvil Publishing, 2012, ISBN 978-80-87270-42-4.
- [11] Khan, I.: What Is MVC and Why Do We Use MVC? [Online; navštíveno 26.1.2017].
URL <http://www.c-sharpcorner.com/article/what-is-mvc-and-why-we-use-mvc/>
- [12] Kononenko, K.: Model-View-Controller (MVC) Explained Through Ordering Drinks At The Bar. [Online; navštíveno 26.1.2017].
URL <https://medium.freecodecamp.com/model-view-controller-mvc-explained-through-ordering-drinks-at-the-bar-efcba6255053#hufaexnvq>
- [13] Korotya, E.: 5 Best JavaScript Frameworks in 2017. [Online; navštíveno 24.4.2017].
URL <https://hackernoon.com/5-best-javascript-frameworks-in-2017-7a63b3870282>
- [14] Krug, S.: *Don't Make Me Think: A Common Sense Approach to Web Usability, 2nd Edition*. New Riders Publishing, 2006, ISBN 0-321-34475-8.
- [15] Liu, D. H.: Need to complete. [Online; navštíveno 28.1.2017].
URL <http://larifari.org/blog/need-to-complete/>
- [16] Nicolaisen, K.: Popularity of JavaScript Frameworks and Libraries. [Online; navštíveno 24.4.2017].
URL <http://blog.honeyiot.io/popularity-of-javascript-frameworks-and-libraries/>
- [17] programmingisterrible.com: Write code that is easy to delete, not easy to extend. [Online; navštíveno 25.1.2017].
URL <http://programmingisterrible.com/post/139222674273/write-code-that-is-easy-to-delete-not-easy-to>
- [18] Sinek, S.: *Start with Why: How Great Leaders Inspire Everyone to Take Action*. Portfolio, 2011, ISBN 978-1591846444.
- [19] usability.gov: User Interface Design Basics. [Online; navštíveno 24.1.2017].
URL <https://www.usability.gov/what-and-why/user-interface-design.html>
- [20] Weinschenk, S.: *100 věcí, které by měl každý designér vědět o lidech*. Computer Press, 2012, ISBN 978-80-251-3649-2.
- [21] Yu-kai Chou: *Actionable Gamification: Beyond Points, Badges, and Leaderboards*. Octalysis Group, 2015, ISBN 1511744049.

Přílohy

Příloha A

Obsah přiloženého DVD

- `technicka_zprava` – Technická zpráva ve formátu PDF
- `video` – Prezentační video
- `poster` – Plakát aplikace S3PS
- `s3ps` – Adresář se zdrojovými soubory aplikace
- `tz` – Adresář se zdrojovými soubory této technické zprávy

Příloha B

Plakát

S3PS

PŘEDSTAV SI,
že za sebou máš
samé **úspěšné**
on-line projekty.

Pá 6.1. Cíl	Spolupráce klienta a designéra	✓ SPLNĚNO
Po 13.2. Cíl	Návrh uživatelského rozhraní	✓ SPLNĚNO
St 22.3. Cíl	Motivace klienta	✓ SPLNĚNO
Čt 6.4. Použito	Přidané prvky gamifikace	✓ SPLNĚNO
Po 14.8. Plán	Používat na reálných projektech	○ SPLNĚNO

Vím, že takové projekty nevznikají přes noc. Neustále se vylepšují a vyvíjí. Vývoj musíš chtít ty sám i tvůj klient. Proto vzniklo **S3PS**, aby se vývoj stal přirozený.

CO JE S3PS?

Webová aplikace pro spolupráci klienta s designérem, která **motivuje ke kontinuální práci na projektu**. Za cíl si klade vytvořit u klienta návyk k jejímu pravidelnému používání a poté jej vybízí k dalšímu vývoji projektu. Uživatelské testování ukázalo, že **S3PS** pomáhá lépe pochopit proces práce na projektu a vyzývá klienta k dalšímu vývoji projektu.

VEDOUCÍ
Ing. Vítězslav Beran, Ph.D.

KONTAKT
zdenek@creepy.cz

AUTOR
Zdeněk Studený



Obrázek B.1: Plakát prezentující zhotovenou aplikaci S3PS.