



FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

KATEDRA INFORMATIKY
A VÝPOČETNÍ TECHNIKY



Bakalářská práce

Přehled GUI systémů pro tvorbu aplikací s minimální potřebou psaní programového textu

Jakub Homolka





FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

KATEDRA INFORMATIKY
A VÝPOČETNÍ TECHNIKY

Bakalářská práce

Přehled GUI systémů pro tvorbu aplikací s minimální potřebou psaní programového textu

Jakub Homolka

Vedoucí práce

Ing. Richard Lipka, Ph.D.

© Jakub Homolka, 2024.

Všechna práva vyhrazena. Žádná část tohoto dokumentu nesmí být reprodukována ani rozšiřována jakoukoli formou, elektronicky či mechanicky, fotokopírováním, nahráváním nebo jiným způsobem, nebo uložena v systému pro ukládání a vyhledávání informací bez písemného souhlasu držitelů autorských práv.

Citace v seznamu literatury:

HOMOLKA, Jakub. *Přehled GUI systémů pro tvorbu aplikací s minimální potřebou psaní programového textu*. Plzeň, 2024. Bakalářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce Ing. Richard Lipka, Ph.D.

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jakub HOMOLKA**
Osobní číslo: **A21B0616P**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační systémy**
Téma práce: **Přehled GUI systémů pro tvorbu aplikací s minimální potřebou psaní programového textu**
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s existujícími nástroji pro automatizovanou tvorbu aplikací s minimální potřebou psaní programového textu.
2. Seznamte se s existující literaturou zaměřenou na UX komplexních nástrojů pro tvorbu aplikací.
3. Srovnajte nalezené nástroje podle dostupnosti a poskytované funkcionality. Vyberte tři nástroje pro podrobnější analýzu.
4. Připravte ukázkový příklad pro porovnání vlastností vybraných nástrojů.
5. Porovnejte použitelnost a UX vybraných nástrojů na základě připravené ukázky. Vyhodnoťte jejich přínos pro vývoj SW běžným uživatelem.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Ing. Richard Lipka, Ph.D.**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **2. října 2023**
Termín odevzdání bakalářské práce: **2. května 2024**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 25. října 2023

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného akademického titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Západočeská univerzita v Plzni má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

Plzeň dne 30. dubna 2024

.....

Jakub Homolka

V textu jsou použity názvy produktů, technologií, služeb, aplikací, společností apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Abstrakt

Tato bakalářská práce zkoumá no-code a low-code platformy, které umožňují automatizovanou tvorbu aplikací s minimálním potřebou psaní programového textu. Práce se zaměřuje na to, jak tyto platformy urychlují vývoj softwaru a zpřístupňují tvorbu aplikací pro nevyvojáře. V úvodu práce jsou představeny existující nástroje a jejich klíčové vlastnosti. Následuje srovnání vybraných platform podle dostupnosti a funkčnosti, z nichž byly vybrány nástroje Adalo, Glide a Softr. Ve všech třech nástrojích byla vytvořena ukázková aplikace správy projektů a následně bylo provedeno jejich celkové zhodnocení. Výsledkem práce je, že nástroje Adalo a Glide jsou velmi přínosné pro běžného uživatele, Adalo pro svou jednoduchou tvorbu databáze a flexibilní tvorbu designu a Glide pro rychlost a jednoduchost transformovat data do funkční aplikace. Naproti tomu nástroj Softr je přínosný pouze částečně, kvůli chybějícímu nastavení viditelnosti komponent a problémům v připojení k Airtable.

Abstract

This bachelor's thesis explores no-code and low-code platforms that enable automated application creation with a minimum need for programming text writing. The work focuses on how these platforms accelerate software development and make application creation available to non-developers. In the introduction to the work, the existing tools and their key characteristics are presented. The following is a comparison of selected platforms based on availability and functionality, from which the tools Adalo, Glide, and Softr were selected. In all three tools, a demonstration application for project management was created and an overall evaluation was subsequently made. As a result of the work, the Adalo and Glide tools are very beneficial for the average user, Adalo for its simple database creation and flexible design creation, and the Glide for the speed and simplicity of transforming data into a functional application. In contrast, the Softr tool is only partially beneficial, due to the lack of component visibility settings and problems in connecting to the Airtable.

Klíčová slova

GUI systémy • UX • automatizovaná tvorba aplikací • no-code • low-code

Poděkování

Rád bych poděkoval Ing. Richardu Lipkovi, Ph.D za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování bakalářské práce. Dále bych rád poděkoval své rodině za neochvějnou podporu a motivaci, kterou mi poskytovala po celou dobu mého studia.

Obsah

1	Úvod	3
2	Nástroje pro tvorbu aplikací s minimální potřebou psaní kódu	5
2.1	Low-code platformy	5
2.2	No-code platformy	5
2.3	Seznam nástrojů	5
2.3.1	Adalo	6
2.3.2	Bubble	8
2.3.3	Glide	10
2.3.4	Google AppSheet	12
2.3.5	Mendix	14
2.3.6	Microsoft Power Apps	16
2.3.7	OutSystems	18
2.3.8	Quickbase	20
2.3.9	Softr	22
2.3.10	Zoho Creator	24
2.4	Tabulka nalezených nástrojů	27
2.5	Umělá inteligence	28
2.5.1	ChatGPT	28
2.5.2	GitHub Copilot	29
2.5.3	Zhodnocení AI jako nástroje na tvorbu aplikací	30
3	UX nástrojů pro tvorbu aplikací	31
3.1	Teorie UX	31
3.2	Modely a metody hodnocení UX	32
3.2.1	Nielsenovo deset heuristik	33
3.2.2	UX Honeycomb od Petera Morvilla	34
3.3	UX v nástrojích pro tvorbu aplikací	35

4 Srovnání nalezených nástrojů	37
4.1 Srovnání no-code platforem	37
4.1.1 Srovnání podle dostupnosti a ceny	37
4.1.2 Srovnání podle funkcionality bezplatných verzí	39
4.2 Srovnání low-code platforem	40
4.2.1 Srovnání podle dostupnosti a ceny	41
4.2.2 Srovnání podle funkcionality bezplatných verzí	42
4.3 Výběr tří nástrojů pro podrobnější analýzu	43
5 Ukázková aplikace pro porovnání nástrojů	45
5.1 Popis ukázkové aplikace	45
5.2 Databáze aplikace	46
5.2.1 Relace	46
5.3 Use-case	48
6 Porovnání použitelnosti a UX vybraných nástrojů	51
6.1 Popis vytváření ukázkové aplikace ve vybraných nástrojích	51
6.1.1 Vývoj databáze	52
6.1.2 Vývoj uživatelského rozhraní (UI) a designu	58
6.1.3 Nasazení	69
6.2 Porovnání nástrojů na základě připravené ukázky	70
6.3 Srovnání UX podle Honeycomb	72
6.4 Vyhodnocení přínosu vybraných nástrojů pro vývoj SW běžným uživatelé	75
6.4.1 Adalo	75
6.4.2 Glide	76
6.4.3 Softr	76
6.4.4 Vyhodnocení	77
7 Závěr	79
Přehled zkratk	81
Bibliografie	83
A Vytvořené aplikace	85
B Obsah přílohy	87
Seznam obrázků	89
Seznam tabulek	91

V současné době dochází k prudkému nárůstu poptávky po efektivních řešeních v oboru technologií a softwarového vývoje, které více zpřístupňují, usnadňují a urychlují tvorbu aplikací. Rostoucí oblíbenost nástrojů pro automatizovanou tvorbu aplikací s minimálním psaním programového textu, známých jako no-code a low-code platformy, je zvláště patrná v tomto trendu. Tyto platformy poskytují revoluční přístup k vývoji softwaru tím, že nabízí grafická uživatelská rozhraní (GUI). GUI těchto platforem poskytují vizuální nástroje, jako jsou například „táhni a pusť“ (drag and drop) konstruktory, přednastavené widgety a formulářové generátory, což uživatelům umožňuje sestavovat komplexní aplikace bez nutnosti psaní kódu. Díky tomu se výrazně snižuje složitost tvorby aplikací a usnadňuje demokratizace procesu vývoje softwaru. Umožňují například rychlé sestavení obchodních aplikací, webových stránek nebo dokonce komplexních informačních systémů bez hlubokých technických znalostí.

Bakalářská práce se zaměřuje na komplexní přehled a analýzu no-code a low-code platforem, které otevírají dveře k automatizované tvorbě aplikací. Primární motivací je prozkoumat, jak tyto nástroje mění klasické pojetí softwarového vývoje a jaký mají potenciál zefektivnit a zjednodušit tvorbu aplikací, přičemž zohlední i potenciální výzvy a omezení, jako je například omezená flexibilita, design, funkčnost nebo závislost na konkrétním nástroji. Úvodní kapitola poskytuje kontext pro další výzkum, představuje se zde rozsáhlá paleta existujících nástrojů, popisující jejich funkcionality a vlastnosti. Hodnocení dostupnosti těchto platforem, možnosti integrace a výkonnosti se stanou základními pilíři pro srovnávací výzkum.

Vybrané nástroje budou podrobeny důkladnému srovnání, kde na příkladu konkrétní aplikace budou testovány jejich vlastnosti, schopnosti a omezení. Tento přístup umožní identifikovat klíčové faktory, které ovlivňují výběr no-code a low-code platforem pro konkrétní typy projektů. Porovnání UX vybraných nástrojů poskytne hlubší porozumění tomu, jak intuitivnost a přístupnost rozhraní podporuje nebo omezuje tvůrčí proces vytváření softwaru pro různé uživatele. Závěrem práce bude vyhodnocení výhod a omezení těchto platforem a jejich přínos pro vývoj softwaru z pohledu běžného uživatele.

Tato práce slibuje nabídnout přínos jak pro akademickou komunitu, tak pro praxi, poskytující hodnotné poznatky pro budoucí směřování no-code a low-code platforem a jejich role v moderním softwarovém inženýrství. Zkoumání této tematiky je aktuální a relevantní, vzhledem k rostoucímu trendu digitalizace a potřebě efektivního softwarového vývoje.

Nástroje pro tvorbu aplikací s minimální potřebou psaní kódu

2

Hlavním úkolem těchto nástrojů je prolomit technologické bariéry a nabídnout přístup k vývoji aplikací bez ohledu na úroveň kódovacích dovedností uživatele. Navíc, přesunutím stylu vývoje od ručního programování s tradičními programovacími jazyky ke grafické interakci uživatelského rozhraní s předem nastavenými komponentami lze proces vývoje aplikace výrazně urychlit a tím i snížit náklady. Cílem je identifikovat, analyzovat a porovnat tyto nástroje, které umožní vytváření aplikace bez toho, aniž by uživatel musel mít jakékoliv znalosti programování psát rozsáhlé kusy kódu.

2.1 Low-code platformy

Low-code platformy poskytují prostředí, které umožňuje uživatelům vytvářet aplikace v grafickém rozhraní s nízkou potřebou psaní kódu. Jsou navrženy pro rychlejší vývoj aplikací a jsou vhodné pro běžné uživatele i vývojáře, kteří mají omezené programátorské dovednosti nebo chtějí urychlit vývojový proces [1].

2.2 No-code platformy

No-code platformy jdou ještě dál ve snaze odstranit nutnost psát kód a zaměřují se na vizuální prostředí a interaktivní nástroje, které umožňují uživatelům vytvářet aplikace pomocí pouhého klikání, táhnutí a nastavování parametrů. Tyto platformy jsou ideální pro uživatele bez jakýchkoliv programátorských znalostí.

2.3 Seznam nástrojů

V následujících podkapitolách se nachází popis nalezených nástrojů na současném trhu. Výběr nástrojů pro tvorbu aplikací bez nutnosti kódování byl proveden sys-

tematickým prohledáváním internetu a odborných zdrojů. Nejdříve byly použity široké vyhledávací dotazy, jako jsou „nejlepší no-code nástroje“, „nejlepší low-code nástroje“ a „jak vytvořit aplikaci bez programování“, což umožnilo získat přehled o oblíbených platformách. Poté bylo vyhledávání zpřesněno specifickými dotazy a hledání na technologických blozích, IT portálech jako TechTarget a G2, a přímo na webových stránkách poskytovatelů. Tyto kroky pomohly sestavit seznam relevantních nástrojů, které budou následně analyzovány a srovnány podle klíčových kritérií, jako jsou dostupnost, cena, funkcionalita a snadnost použití.

2.3.1 Adalo

Web	www.adalo.com
Licence	Freemium, Starter 45 USD za měsíc, Professional 65 USD za měsíc, Team 200 USD za měsíc a Enterprise 250 USD za měsíc
Datum poslední verze	1.12. 2022
Výrobce	Adalo, Inc.
Typ vstupu	vizuální modelování, integrace databáze, logické a podmíněné vstupy, API a externí integrace, nastavování vlastností
Výstupní technologie	Webové aplikace, Mobilní aplikace, Progresivní web aplikace (PWA)
Cílové platformy	Web, iOS, Android

Tabulka 2.1: Informace o platformě Adalo

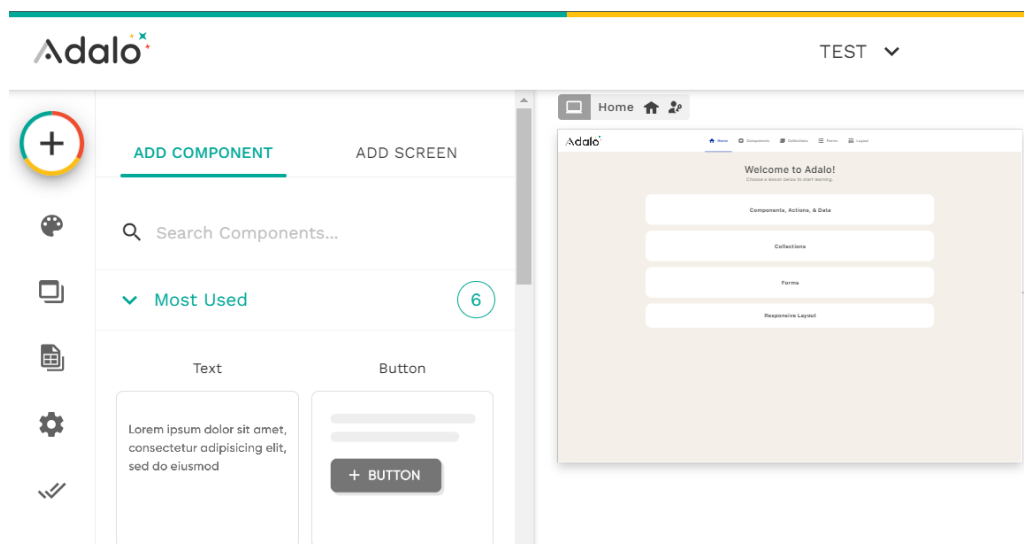
Adalo je no-code platforma, založená v roce 2018 Davidem Adkinem, která umožňuje uživatelům vytvářet mobilní a desktopové aplikace bez nutnosti psaní kódu, včetně Progressive Web App (PWA) [2]. PWA mohou nabízet funkce jako instalace na domovskou obrazovku, push notifikace a možnost offline práce, což je činí podobnými nativním aplikacím, zatímco klasické webové aplikace tyto funkce standardně neposkytují. S Adalo lze vytvořit většinu typů aplikací pomocí grafického rozhraní, které usnadňuje navrhování a vytváření aplikace bez jakýchkoliv technických dovedností.

Uživatelské rozhraní je navrženo s ohledem na jednoduchost a přístupnost, což poskytuje uživatelům bezproblémovou práci s funkcí „drag and drop“. Tato funkce velice zjednodušuje tvorbu uživatelského rozhraní aplikace díky snadnému přetahování komponentů na obrazovku. Funkcionalita, vlastnosti a vzhled jednotlivých komponent lze nastavit prostřednictvím široké škály dostupných možností, které komponenty nabízejí. Tento způsob konfigurace, často označovaný jako „vyber a klikni“ (point and click), poskytuje intuitivní a efektivní způsob, jak upravovat

aplikace bez nutnosti psaní kódu. Tímto způsobem lze rychle vytvářet a upravovat uživatelské rozhraní podle konkrétních požadavků. Celkově Adalo klade velký důraz na to, aby uživatelské rozhraní editoru bylo přehledné, efektivní, snadno použitelné a atraktivní pro uživatele. Pro inspiraci a urychlení procesu vývoje aplikace má Adalo rozsáhlý soubor šablon a motivů, které lze jednoduše upravit podle konkrétních potřeb.

Adalo nabízí více než 50 komponentů uživatelského rozhraní, které většina aplikací vyžaduje, jako jsou např. tlačítka, text, seznamy, obrázky, formuláře a navigační panely. Dalším důležitým prvkem je široká škála předem připravených dynamických akcí pro interaktivitu aplikace. Uživatelé mohou snadno implementovat funkce, jako je otevírání oken či stránek, odesílání e-mailů, odesílání požadavků HTTP a přístup k funkcím zařízení, jako je fotoaparát. Uživatelé mohou také přidávat vlastní akce, pokud mají znalosti programování, které jim dovoluje rozšířit funkčnost aplikace za hranice předem sestavených akcí.

Správa dat v Adalo je velice snadná díky své vestavěné databázi, která umožňuje vytvářet kolekce, nastavovat vztahy a bezpečnostní pravidla pro kontrolu přístupu k datům. Kromě toho povoluje integrační možnost s existujícími databázemi a tím dává možnost využít Adalo jako frontend pro aplikaci. Pokud chcete přidat funkce, jako je přihlášení, registrace, resetování hesla a ověření e-mailu, systém správy uživatelů vám umožní snadno přidat ověření do aplikace.



Obrázek 2.1: Uživatelské rozhraní Adalo

Adalo disponuje rozsáhlými možnostmi pro integrování s externími službami, API a pluginy, které umožňují uživatelům propojit své aplikace s širokým spektrem

externích systémů a služeb. Prostřednictvím nástroje Zapier¹, který přináší integraci s více než 6000 službami, jako jsou CRM systémy, e-mailové platformy, sociální média a mnoho dalších, Adalo zvětšuje svou flexibilitu a funkčnost.

Adalo je efektivní pro vytváření prototypů a menších až středních aplikací. Pro velké, vysoce škálovatelné aplikace s vysokou mírou uživatelské interakce může Adalo narazit na omezení v oblasti výkonu a optimalizace.

Díky aktivní komunitě uživatelů lze snadno sdílet zkušenosti a poskytovat vzájemnou podporu. Navíc Adalo má k dispozici školení a technickou podporu pro vývojáře, a tím jim pomáhá vytvářet aplikace s větší jistotou a efektivitou.

S Adalo můžete snadno publikovat vaši aplikaci na Google Play, Apple App Store, kde je ale zapotřebí mít Apple vývojářskou licenci, a na web. Adalo vygeneruje potřebné soubory a stačí je odeslat do příslušného obchodu. Aplikaci lze také publikovat jako PWA.

Adalo nabízí jeden bezplatný plán, který slouží k vyzkoušení nástroje a jeho funkcí. Pokud potřebujete více funkcí, než má bezplatný plán, existují 4 placené plány s rozšířenými funkcemi a podporou, kterou potřebujete.

2.3.2 Bubble

Web	www.bubble.io
Licence	Freemium, Starter 32 USD za měsíc, Growth 134 USD za měsíc, Team 399 USD za měsíc a Enterprise s nastavitelnou cenou
Datum poslední verze	17.10. 2023
Výrobce	Bubble Group, Inc.
Typ vstupu	vizuální modelování, integrace databáze, logické a podmíněné vstupy, API a externí integrace, nastavování vlastností, programový kód
Výstupní technologie	Webové aplikace, PWA
Cílové platformy	Web

Tabulka 2.2: Informace o platformě Bubble

Vizuální programovací platforma založená v roce 2012 společností Bubble Group, Inc. Jedná se o no-code platformu zaměřující se na vývoj webových aplikací [3].

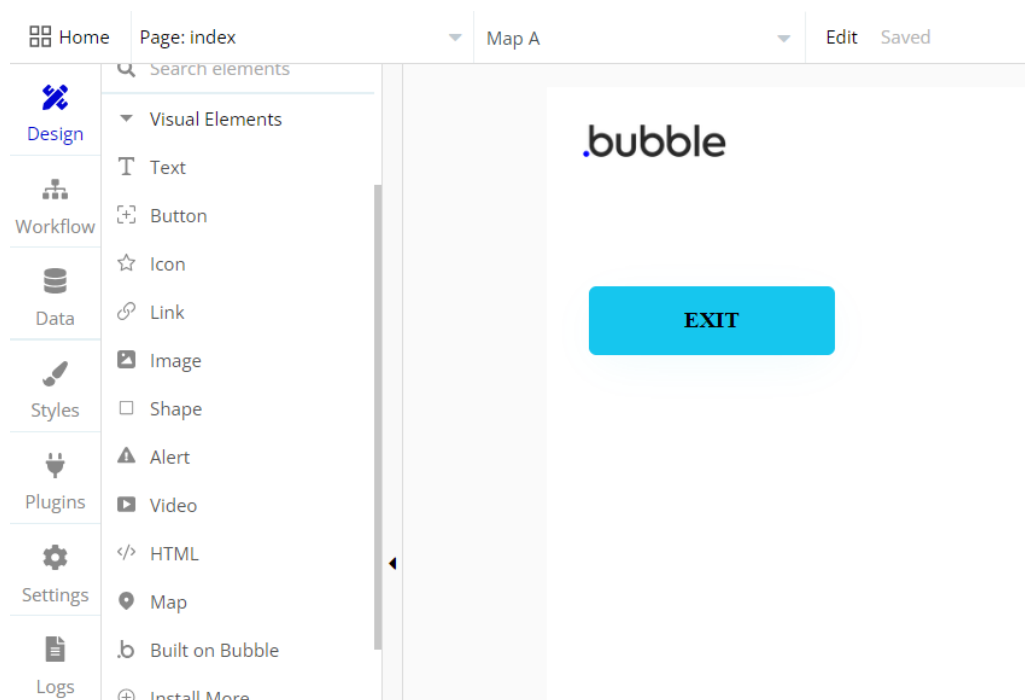
Uživatelské rozhraní platformy je přehledné a uživatelsky příjemné. V Bubble se využívá vizuálního programování a grafického rozhraní, kde uživatelé propojují vizuální prvky pro definici funkcí a interakcí. Prostředí umožňuje snadný návrh

¹Kompletní seznam integrací, které Zapier nabízí, najdete na oficiálních stránkách Zapier: <https://zapier.com/apps>

a vývoj interaktivní aplikace pro desktopové a mobilní webové prohlížeče. Šablony v Bubble jsou předem vytvořené webové aplikace nebo projekty, které mohou sloužit jako výchozí bod pro tvorbu vlastních aplikací. Příkladem takových šablon jsou například Marketplaci, Dashboardy, Správci úkolů atd. Tyto šablony obsahují předdefinované prvky, designy a funkce, které lze upravit a přizpůsobit podle konkrétních požadavků uživatele.

Bubble obsahuje různé vizuální komponenty pro tvorbu uživatelského rozhraní. Tlačítka, textová pole, obrázky a další prvky jsou k dispozici v knihovně, což poskytuje možnost snadného designování a uspořádání stránek. Uživatelé mohou definovat různé akce a události pomocí vizuálních bloků, jako je otevírání oken, posílání e-mailů, odeslání formuláře, časové spouštěče, aktualizace dat a mnoho dalších interakcí. Přidání podmínek k různým komponentám, které se nastavují výběrem stavů nebo vlastností a výběrem logiky, a logických operací do svých aplikací dovoluje vytvářet dynamické a responzivní chování aplikace na základě různých scénářů.

Bubble dává k dispozici bezpečné a škálovatelné úložiště dat, kde se dají navrhovat a spravovat datové struktury. Vestavěný systém správy uživatelů usnadňuje nastavení uživatelský účtů a přihlášení pomocí hesel nebo jakéhokoliv poskytovatele kompatibilního s OAuth 2.0, jako je Google nebo Facebook.



Obrázek 2.2: Uživatelské rozhraní Bubble

Aplikace vytvořené v Bubble.io lze rozšířit pomocí pluginů, které mohou po-

skytovat různé funkce, grafické prvky, komunikační nástroje a služby třetích stran. Bubble má vlastní trh s pluginy, kde uživatelé mohou vyhledávat správná rozšíření, která potřebují. Uživatelé mohou také vytvářet vlastní pluginy a tím přidat specifickou funkcionalitu do své aplikace. Bubble.io využívá také integrace s externími systémy a službami pro jejich vzájemnou spolupráci a komunikaci, která zahrnuje získávání, ukládání a výměnu informací jako např. Zapier, Slack, Google Maps apod.

Bubble je sestaveno tak, aby bylo schopné zvládnout projekty různých velikostí. Bez ohledu na to, zda vytváříte menší webovou stránku nebo komplexní webovou aplikaci, platforma přináší prostředky pro škálování aplikace podle potřeb uživatele. Bubble je postaven na cloudové infrastruktuře, což znamená, že aplikace hostované na této platformě mohou být automaticky škálovány. To zahrnuje možnosti zvýšení výpočetního výkonu, paměti a úložného prostoru, když to situace vyžaduje. Uživatelé mohou vybírat z různých plánů předplatného, které se liší ve výkonnosti a kapacitě, a podle potřeby mohou přecházet na vyšší plány. Platforma Bubble dále poskytuje uživatelům nástroje, které jim umožňují sledovat, jak dobře jejich aplikace funguje. S těmito nástroji mohou měřit, jak rychle se aplikace načítá a reaguje na uživatelské vstupy. Díky tomu je možné snadno zjistit, kde mohou být v aplikaci zpoždění nebo technické problémy a umožňuje se na tyto problémy zaměřit a vyřešit je.

Kolem platformy existuje aktivní komunita uživatelů, kteří poskytují podporu na fórech, kde sdílí své poznatky a zkušenosti s nástrojem. Pokud by podpora komunity nestačila, Bubble má také vlastní technickou podporu a uživatelé si mohou přečíst technickou dokumentaci. Pro začátečníky nabízí platforma různá výuková videa a také placené výukové tábory.

Aplikace vytvořená v Bubble lze publikovat na webových serverech společnosti.

V nabídce Bubble jsou čtyři placené balíčky, kde u tří z nich jsou stanoveny ceny s měsíčním nebo ročním předplatným a obsah balíčku. Čtvrtý si může uživatel vybrat, co chce do balíčku zahrnout a cena bude stanovena podle zvolených funkcí. Poslední pátý balíček je bezplatný, který slouží primárně pro seznámení se s nástrojem.

2.3.3 **Glide**

Inovativní no-code platforma, která umožňuje uživatelům vytvářet PWA a webové aplikace bez nutnosti hlubších znalostí programování [4]. Tato platforma je zaměřena především na tvorbu aplikací pro zpracování a organizaci dat, jako jsou například seznamy a knihovny.

Platforma je vysoce vizuální a používá jednoduché rozhraní s mnoha komponentami, které se řadí za sebe do tzv. kontejnerů. Tyto kontejnery se pak ihned vizualizují v náhledu aplikace. Glide nabízí více než 400 šablon pro zrychlení procesu

Web	www.glideapps.com
Licence	Freemium, Maker 49 USD za měsíc, Team 99 USD za měsíc, Business 249 USD za měsíc a Enterprise 1 000 USD za měsíc
Datum poslední verze	1.12. 2023
Výrobce	Glide Labs, Inc.
Typ vstupu	data z tabulkových editorů, vizuální modelování, integrace databáze, logické a podmíněné vstupy, API a externí integrace, nastavování vlastností
Výstupní technologie	Webové aplikace, PWA
Cílové platformy	Web, iOS, Android

Tabulka 2.3: Informace o platformě Glide

vytváření aplikace, pro předvedení komponent v aplikaci a pro poskytnutí výchozího bodu, pokud si uživatel není jistý, kde začít. Zatímco jednoduchost a snadné použití jsou silnými stránkami Glide, jeho přístup k přizpůsobení je poněkud omezenější ve srovnání s jinými platformami.

Velkou výhodou Glide je, že dokáže automaticky vytvořit aplikaci z tabulky, protože se integruje se zdroji dat, jako jsou Google Sheets, CSV soubory, tabulky Glide, Airtable nebo Excel. Používání těchto tabulek jako databázi aplikace znamená, že máte plnou kontrolu nad svými daty ve vizuálním rozhraní. Glide nabízí dvoucestnou synchronizaci mezi aplikací a tabulkou. To znamená, že když uživatelé provádějí změny v aplikaci, tyto změny jsou automaticky zaznamenány a aktualizovány v příslušné tabulce. Obdobně, jakékoli změny provedené přímo v tabulkách se projeví v aplikaci.

Glide poskytuje nastavení přístupových práv k jednotlivým datům aplikace. Díky této funkci je jasně definováno, kdo může vidět nebo upravovat určitá data. Tím se sníží riziko neoprávněného přístupu k citlivým informacím.

Glide disponuje vlastními API konektory a nástroji, které usnadňují propojení s externími službami. Tyto konektory mohou zahrnovat integrace s nástroji jako Zapier, Integromat, OpenAI nebo jinými službami. Pro pokročilé uživatele, kteří potřebují specifickou integraci, Glide povoluje možnost přidání pluginů, rozšíření a vytváření vlastních skriptů, které definují vlastní logiku pro integrace.

Platforma je přizpůsobena i pro projekty, které vyžadují vyšší funkcionalitu, s velkým počtem uživatelů. Pro tyto projekty se doporučuje prémiový plán.

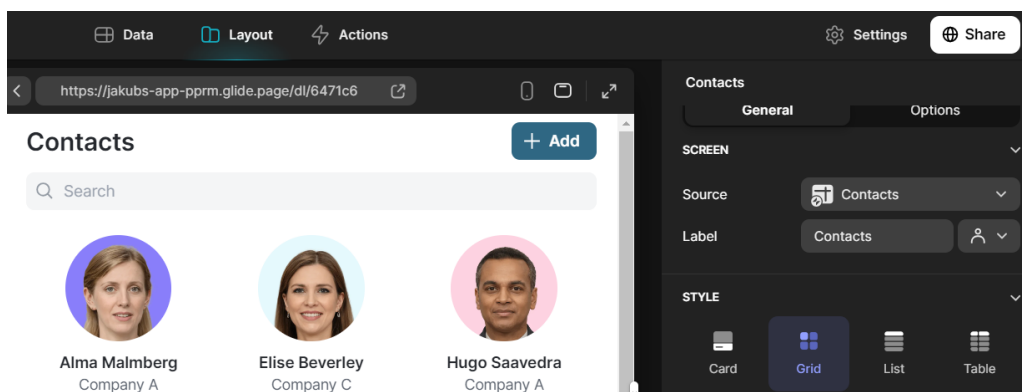
Glide University² je webová stránka, která obsahuje všechny kurzy, lekce a průvodce pro snadnější seznámení se s platformou. Dále se na stránce nachází diskusní

²Odkaz na Glide University: <https://learn.glideapps.com/>

fórum s aktivní komunitou, která může pomoci s jakýmkoliv problémem. Samotný Glide pak přináší pomoc v podobě technické podpory.

Aplikace vytvořené v Glide lze snadno publikovat jako webové aplikace nebo nainstalovat jako mobilní aplikace prostřednictvím generovaných odkazů a QR kódů.

Glide nabízí odstupňovanou cenovou strukturu, od bezplatného plánu pro jednotlivce vytvářející své první aplikace až po cenové plány pro velké podniky, které potřebují pokročilé zabezpečení a ovládací prvky.



Obrázek 2.3: Uživatelské rozhraní Glide

2.3.4 Google AppSheet

AppSheet je platforma založená roku 2014 Praveenem Seshadrim, později byla akvizována společností Google. Tato no-code platforma poskytuje uživatelům možnost vytvářet mobilní a webové aplikace přímo z tabulek a databází, například z Tabulek Google, Excelu nebo jiných databázových systémů [5].

Uživatelské rozhraní této platformy je velmi přívětivé a přehledné a snadno se v něm vytváří a upravují aplikace pomocí přetahování a vkládání prvků. Design UI je optimalizován tak, aby byl snadno použitelný a umožňoval uživatelům přizpůsobit si obsah aplikace podle svých individuálních potřeb. Toto přizpůsobení může zahrnovat změnu rozložení, stylů nebo funkcionality aplikace na základě uživatelských preferencí a interakcí. Uživatelé si mohou vybírat z mnoha komponent a widgetů, které mohou snadno přidat do svých aplikací a jednoduše nastavit jejich vlastnosti. Google AppSheet klade také velký důraz na rychlost, spolehlivost a jednoduchost používání platformy pro nejvyšší pohodlí uživatele při vývoji aplikace. Aplikace vytvořené v této platformě mají responzivní design, který přizpůsobuje zobrazení na různých zařízeních.

Aplikace v AppSheet mohou načítat data např. ve formě obrázků, podpisů, geolokace, čárových kódů a NFC. Data se poté automaticky synchronizují do cloudu

Web	www.appsheet.com
Licence	Freemium, Starter 5 USD na uživatele za měsíc, Core 10 USD na uživatele za měsíc, Enterprise Standard a Enterprise Plus s nastavitelnou cenou
Datum poslední verze	1.12. 2023
Výrobce	Google LLC
Typ vstupu	data z tabulkových editorů, vizuální modelování, integrace databáze, logické a podmíněné vstupy, API a externí integrace, nastavování vlastností
Výstupní technologie	Webové aplikace, Mobilní aplikace, PWA
Cílové platformy	Web, iOS, Android

Tabulka 2.4: Informace o platformě Google AppSheet

nebo je uživatel může synchronizovat data ručně. Zobrazení dat zahrnují tabulky, formuláře, mapy, grafy a kalendáře. Jedna aplikace může mít více zobrazení skládající se z dat z různých zdrojů. Přístup k datům je možný i offline, protože uložení dat se nachází v zařízení a synchronizace se zobrazí až po obnovení připojení k internetu.

Z hlediska zabezpečení má Google AppSheet mnoho bezpečnostních opatření, aby mohl chránit citlivá data a samotnou aplikaci před cizím útokem. Platforma využívá šifrování pro uložená data i data, která jsou v pohybu, což zajišťuje bezpečný přenos a uchování informací. Identifikace a přístupová práva jsou zabezpečeny moderními autentizačními a autorizačními prostředky, včetně jednorázových přihlašovacích kódů a dvoufaktorového ověření.

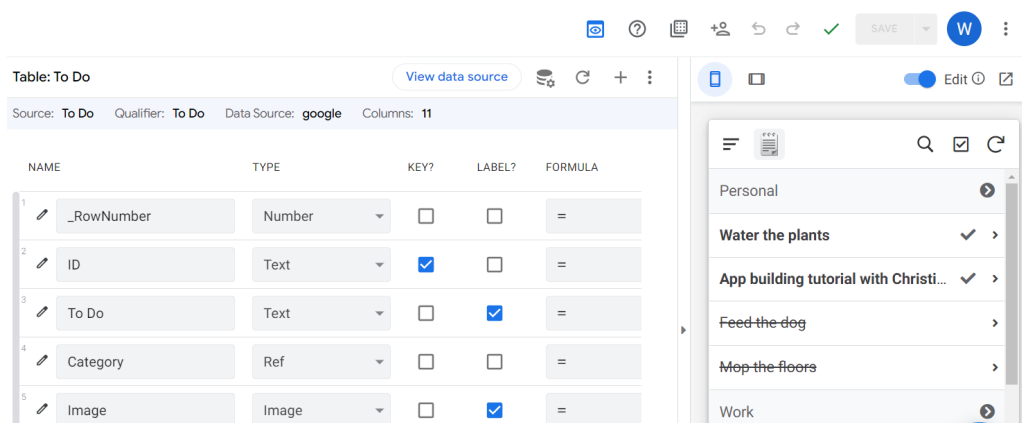
Google AppSheet umožňuje možnost integrace s různými datovými a externími službami. Uživatelé mohou jednoduše propojit své aplikace například s tabulkami Google Sheets, které umožňují jednoduchý import a aktualizaci dat přímo ze známého tabulkového prostředí. Navíc, prostřednictvím knihovny pluginů, mohou uživatelé rozšířit funkčnost svých aplikací podle specifických potřeb.

Platforma nabízí uspokojivou škálovatelnost a výkon díky využívání cloudových technologií. Kapacita se automaticky mění svou velikostí podle potřeby, což se hodí při rostoucím počtu dat v databázi. Takovou menší nevýhodou je závislost na rychlosti a kvalitě připojení k datovým zdrojům.

Komunita okolo Google AppSheet je velmi aktivní v AppSheet fóru i na sociálních sítích, kde uživatelé mohou sdílet své zkušenosti nebo řešit své problémy s platformou. Při nevyřešení problému skrz komunitu se může uživatel obrátit na technickou podporu. Google AppSheet dále poskytuje podrobnou technickou dokumentaci, online semináře, workshopy a další školicí materiály.

Aplikace vytvořené v Google AppSheet lze publikovat různými způsoby, napří-

2 Nástroje pro tvorbu aplikací s minimální potřebou psaní kódu



Obrázek 2.4: Uživatelské rozhraní Google AppSheet

klad přes platformy Google Play Store a Apple App Store, nebo prostřednictvím webu a veřejných odkazů.

V nabídce Google AppSheet je celkem 5 plánů: jeden bezplatný plán, placené plány Starter a Core, s fixní cenou, a plány Enterprise a Enterprise Plus, u kterých si uživatel musí vyžádat cenovou nabídku.

2.3.5 Mendix

Web	www.mendix.com
Licence	Freemium, Basic od 52,50 EUR za měsíc, Standard od 900 EUR za měsíc a Premium s nastavitelnou cenou
Datum poslední verze	28.6. 2023
Výrobce	Siemens AG
Typ vstupu	vizuální modelování, integrace databáze, logické a podmíněné vstupy, API a externí integrace, nastavování vlastností, programový kód
Výstupní technologie	Webové aplikace, Mobilní aplikace, PWA
Cílové platformy	Web, iOS, Android

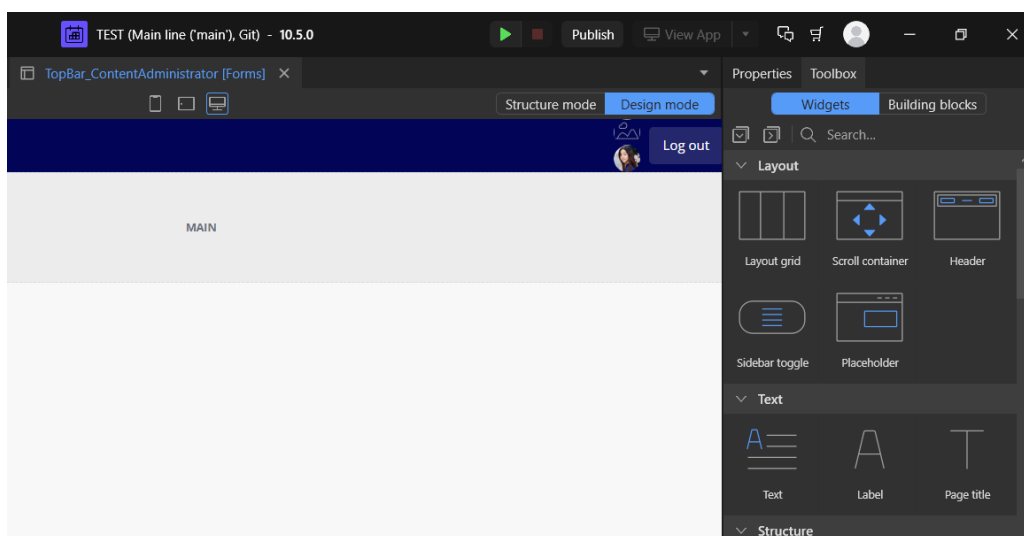
Tabulka 2.5: Informace o platformě Mendix

Mendix je low-code platforma pro vývoj mobilních a webových aplikací, která byla založena v roce 2005 [6]. Spoluzakladateli platformy jsou Derek Roos, Rold Kruit a Deckjan Kruit. Platforma se zaměřuje na usnadnění a zrychlení vývoje podnikových aplikací, které mají vysokou funkčnost, prostřednictvím vizuálního

programování. Aplikace vytvořené v Mendix pak slouží jako nástroje pro efektivní spolupráci uvnitř podniku.

Prostřednictvím rozsáhlé sady vizuálních modelů se mohou vývojáři s různými úrovněmi dovedností podílet na vývoji aplikace. Platforma dříve nabízela dvě vývojová prostředí, které v roce 2023 sloučila dohromady do jednoho IDE Mendix Studio Pro, které je zaměřené na vývojáře s různými úrovněmi dovedností a pozadími, aby mohli spolupracovat s obchodními zainteresovanými stranami na vytváření aplikací rychleji a efektivněji. Zkušenější vývojáři mohou rozšířit funkčnost své aplikace pomocí Javy a JavaScriptu. Mendix Studio Pro navíc umožňuje vývoj řízený modelem, který používá vizuální modely pro definování aplikační logiky, procesních toků a uživatelského rozhraní. Toto prostředí poskytuje komplexní, ale přehledné uživatelské rozhraní s velkým výběrem komponent, widgetů a také šablon, které zrychlují a zlehčují vývoj designu aplikace.

Mendix poskytuje velký výběr funkcionalit, včetně vizualizace dat, workflow, integrace s externími službami a API. Mendix usnadňuje integraci s API pomocí API konektorů a ve svém Mendix Marketplace nabízí mnoho pluginů a rozšíření, které mohou obohatit aplikaci o další funkcionality.



Obrázek 2.5: Uživatelské rozhraní Mendix

Mendix Studio Pro nabízí integraci s Týmovým serverem, který využívá systém správy verzí založený na Gitu. Díky tomu mohou členové týmu efektivně spolupracovat na vývoji aplikací. Tento nástroj umožňuje uživatelům snadno sdílet změny, sledovat historii revizí a koordinovat úkoly mezi různými vývojáři, což zjednodušuje správu a aktualizace projektů v dynamickém a kolaborativním prostředí.

Mendix má svůj nástroj na ukládání a správu dat nazývaný Mendix Data Storage, k tomu také umožňuje připojení s mnoha externími databázemi. Platforma zajišťuje bezpečnost vytvořených aplikací pomocí nástrojů na ověřování, šifrování a správu přístupu.

Oficiální a dostupná dokumentace a aktivní komunita uživatelů poskytují podporu a sdílení zkušeností pro nové i zkušené uživatele. Platforma také poskytuje školení, webináře a technickou podporu pro vývojáře.

Aplikace lze publikovat na různých platformách, včetně cloudových služeb a mobilních zařízení. Samotnou aplikaci pak lze nasadit na Mendix Cloudu, který se dělí na veřejný, vyhrazený a privátní. Mendix Cloud je služba založená na technologii Cloud Foundry, která zjednodušuje proces nasazení bez závislosti na programovacím jazyku a umožňuje snadné škálování a provoz aplikace.

Mendix nabízí různé cenové modely, včetně Freemium, tří placených modelů a Premium modelu s nastavitelnou cenou. Modely jsou strukturovány podle počtu požadovaných funkcionalit.

2.3.6 Microsoft Power Apps

Microsoft Power Apps je low-code platforma, kterou uvedla společnost Microsoft na konci roku 2015 jako součást své nabídky platformy jako služby [7]. Tato platforma poskytuje sadu aplikací, služeb a konektorů, které pomáhají vývojářům i ne-technickým uživatelům snadno a rychle vyvíjet vlastní obchodní aplikace s minimálními nebo žádnými znalostmi programování. Aplikace vytvořené v Microsoft Power Apps umožňují implementaci komplexní obchodní logiky a automatizaci pracovních postupů. Tím se transformují tradiční manuální obchodní operace do digitální formy, což zvyšuje efektivitu a snižuje možnost chyb v procesech.

Web	www.microsoft.com
Licence	Freemium a Premium 20 USD za měsíc
Datum poslední verze	15.12 2023
Výrobce	Microsoft Corporation
Typ vstupu	vizuální modelování, integrace data z různých zdrojů, logické a podmíněné vstupy, API a externí integrace, nastavování vlastností, programový kód
Výstupní technologie	Webové aplikace, Mobilní aplikace
Cílové platformy	Web, iOS, Android

Tabulka 2.6: Informace o platformě Microsoft Power Apps

Uživatelské rozhraní platformy je přívětivé a pro uživatele Microsoft platformem podobné jako u jiných nástrojů společnosti. Grafické rozhraní obsahuje mnoho různých prvků, komponent, médií, formulářů a widgetů pro konstrukci aplikace. Při

návrhu aplikace Microsoft Power Apps nabízí UI kit, který z designu vytvořeného ve Figma vytvoří rovnou aplikaci. UI kit obsahuje komponenty, vzorové aplikace, informace o nich a jak je použít. Další způsob návrhu aplikace je modelově řízený návrh (model driven design), který vyžaduje kombinaci dovedností v oblasti návrhu databází, uživatelského rozhraní a procesů. Platforma dále poskytuje různé šablony, které mohou proces vývoje urychlit nebo usnadnit. Tyto šablony se nacházejí v galerii šablon a obsahují různé návrhy aplikací, které si uživatel může dále upravit. Jako speciální funkci Power Apps umožňuje tvorbu role-based uživatelského rozhraní, které zajišťuje, že různí uživatelé s různými rolami vidí obsah odpovídající jejich rolím a oprávněním v podniku.

Díky AI Builderu v Power Apps mohou uživatelé do svých aplikací zahrnout umělou inteligenci, aniž by museli hluboce rozumět strojovému učení. Uživatelé mohou automatizovat složité úkoly, jako je analýza textu, zpracování obrázků a predikce, pomocí zavedených modelů umělé inteligence nebo vytvářením vlastních modelů.

Microsoft Power Apps umožňuje uživatelům ukládat data na datové platformě Microsoft Dataverse nebo v jiných zdrojích dat jako je SharePoint, Microsoft 365, Dynamics 365 a SQL Server. Ochranu citlivých dat platforma chrání omezením přístupu a přiřazováním rolím. Integrací s externími službami se může rozšířit správa uživatelů a přidání šifrování dat.

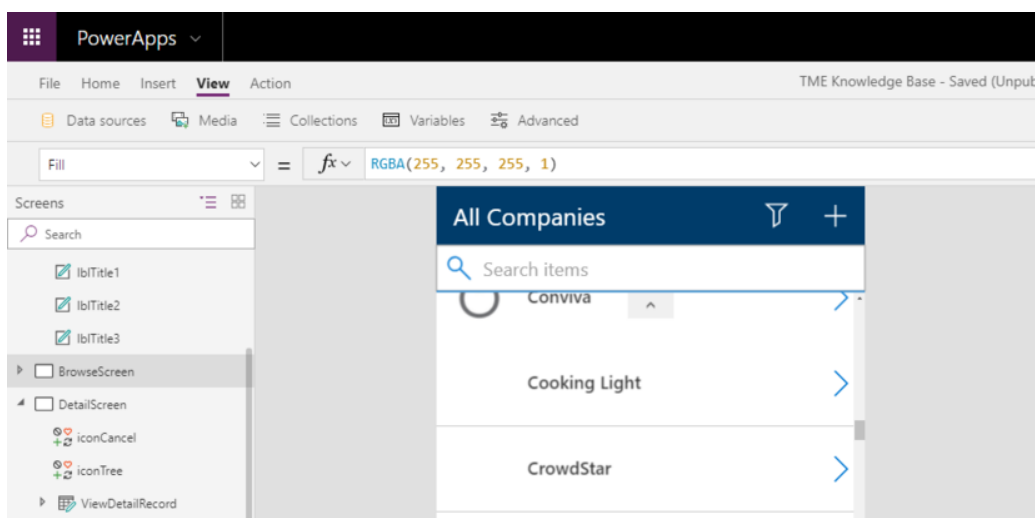
Microsoft Power Apps poskytuje rozsáhlé možnosti integrace externích služeb a API, které rozšíří funkcionalitu podnikové aplikace. Platforma nabízí jak širokou škálu svých předdefinovaných konektorů pro cloudové i lokální služby, tak umožňuje vytvářet vlastní konektory pro propojení s jakýmkoliv API, které podporuje REST nebo OData.

Škálovatelnost aplikace je podpořena efektivním načítáním dat díky využití jmenných vzorců, což zlepšuje výkon při zpracování velkých objemů dat. Jmenné vzorce umožňují aplikaci rychleji a přesněji přistupovat k určeným segmentům dat, což minimalizuje zatížení systému při rostoucím počtu uživatelů nebo dat. Pro rozsáhlejší aplikace je potřeba rozdělit vzorce na menší části a využívat funkce pro rozložení logiky. Aplikaci lze také rozčlenit na jednotlivé sekce pro lepší výkon.

Microsoft Power Apps nabízí uživatelům svou oficiální dokumentaci, podporu, školení a workshopy. Existuje také aktivní komunita uživatelů, kteří sdílejí své zkušenosti a pořádají komunitní aktivity pro uživatele všech dovedností.

Aplikace vytvořené pomocí Microsoft Power Apps se především publikují na platformě Microsoft Power Platform. Při publikaci jsou aplikace zpřístupněny všem uživatelům, se kterými jsou sdíleny. Aplikace je přístupná prostřednictvím webových prohlížečů nebo mobilních zařízení pomocí platformy Microsoft Power Apps.

Microsoft Power Apps nabízí jeden plán pro Power Apps Premium za 20 dolarů měsíčně a jeden bezplatný plán pro jednotlivce, kteří chtějí testovat a vytvářet



Obrázek 2.6: Uživatelské rozhraní Microsoft Power Apps

aplikace pro osobní účely.

2.3.7 OutSystems

Web	www.outsystems.com
Licence	Single app zdarma, Multiple apps od 32 500 USD za rok a Large app portfolio s nastavitelnou cenou
Datum poslední verze	9.10. 2023
Výrobce	OutSystems, S.A.
Typ vstupu	vizuální modelování, integrace databáze, logické a podmíněné vstupy, API a externí integrace, nastavování vlastností, programový kód
Výstupní technologie	Webové aplikace, Mobilní aplikace, PWA
Cílové platformy	Web, iOS, Android

Tabulka 2.7: Informace o platformě OutSystems

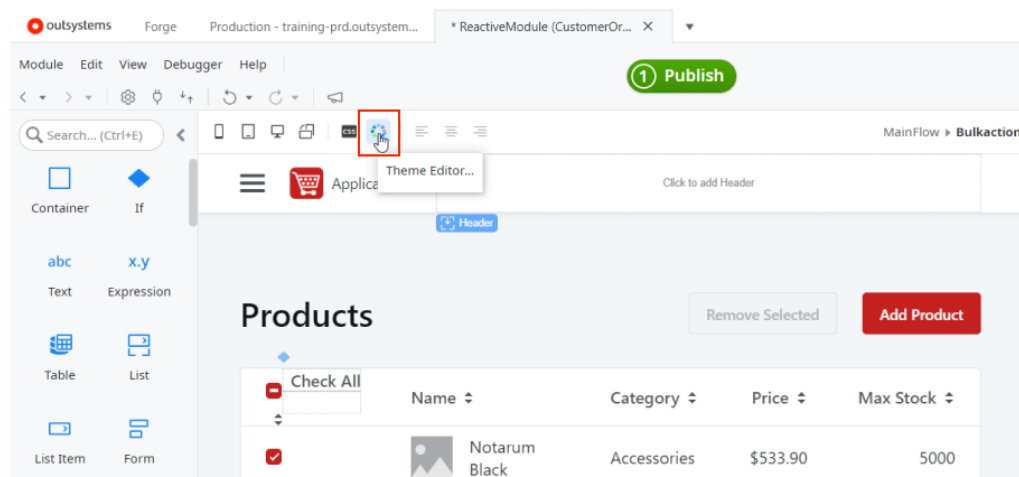
OutSystems je low-code platforma založená v roce 2001 v Portugalsku. Platforma využívá cloudové technologie, výkon AI a vizuální modelování k urychlení procesu vývoje, nasazení a správy omnichannel podnikových aplikací tak, aby vyhovovala potřebám podniku [8]. Poskytuje možnost uživatelům s nízkými nebo žádnými zkušenostmi s programováním podílet se na vývoji aplikací.

Uživatelské rozhraní je sestaveno tak, aby bylo přehledné a intuitivní pro uživatele všech úrovní zkušeností. Vývojové prostředí platformy se dělí na Service Studio

pro samotné vytváření aplikace, Integration Studio pro rozšiřování funkcí pomocí integrace se systémy třetích stran a Platform Server, který obstarává optimalizaci, kompilaci, rozvinutí a nasazení aplikace. Dále platforma dává možnost použít řadu nástrojů pro správu aplikace, jako je Service Center a LifeTime.

V OutSystems se nachází řada předdefinovaných šablon, které slouží uživateli jako výchozí bod pro tvorbu uživatelského rozhraní aplikace. OutSystems také nabízí širokou škálu UI patternů, které lze snadno vybrat a přetáhnout do aplikace. UI patterny zahrnují soubor běžně používaných funkčních a designových prvků uživatelského rozhraní, jako jsou formuláře, tlačítka a navigační prvky. UI patterny a šablony lze snadno přizpůsobit, aby vyhovovaly specifickým potřebám a designu dané aplikace. OutSystems dále podporuje tvorbu responzivních aplikací, které umožňují, aby aplikace automaticky přizpůsobovala svůj vzhled různým velikostem a orientacím obrazovek.

Jako výchozí databázi používá platforma OutSystems MS SQL Server, k němuž dále uživatel může díky integraci využívat různé externí zdroje dat. OutSystems obsahuje nástroje pro modelování dat, které definují strukturu dat a vztahy mezi datovými objekty. Bezpečnost dat v OutSystems je dosažena vestavěnými bezpečnostními funkcemi, které jsou automaticky začleněny do všech aplikací. Hlavními funkcemi jsou šifrování uložených dat, integrace s identity management systémem a přístupová práva založená na rolích.



Obrázek 2.7: Uživatelské rozhraní OutSystems

Platforma OutSystems poskytuje API konektory a jiné nástroje, které umožňují snadné propojení s různými externími systémy. Tyto nástroje zahrnují integrační studio pro vytváření vlastních rozšíření a konektorů.

OutSystems je navržen pro škálovatelnost, proto zvládá rostoucí zatížení aplikace, jako je zvýšený počet uživatelů, transakcí nebo objemu dat. Podporuje vertikální škálovatelnost (zvyšování výpočetní kapacity jednotlivých komponent, jako jsou servery) a horizontální škálovatelnost (přidávání dalších komponent pro distribuci zatížení).

OutSystems dává k dispozici svou oficiální dokumentaci, různé školicí kurzy, technické workshopy, výuková videa a technickou podporu pro své uživatele. Platforma má také aktivní komunitu, kde se sdílejí informace a zkušenosti.

Webové aplikace vytvořené v OutSystems lze publikovat na cloudové službě OutSystems nebo na vlastní serverové infrastruktuře a mobilní aplikace mohou být publikovány na platformách jako je Google Play nebo App Store.

OutSystems spřístupňuje svou platformu zdarma pro začátečníky, kteří si chtějí platformu vyzkoušet a osahat. Cena placené verze začíná od 32 500 USD ročně, dále je tu plán pro velké firmy, kde se cena nastaví podle domluvy.

2.3.8 Quickbase

Web	www.quickbase.com
Licence	30denní zkušební verze, Team od 35 USD za měsíc, Business od 55 USD za měsíc, Enterprise s nastavitelnou cenou
Datum poslední verze	17.12. 2023
Výrobce	Quickbase, Inc.
Typ vstupu	data z tabulkových editorů, vizuální modelování, integrace databáze, logické a podmíněné vstupy, API a externí integrace, nastavování vlastností
Výstupní technologie	Webové aplikace, Mobilní aplikace
Cílové platformy	Web, iOS, Android

Tabulka 2.8: Informace o platformě Quickbase

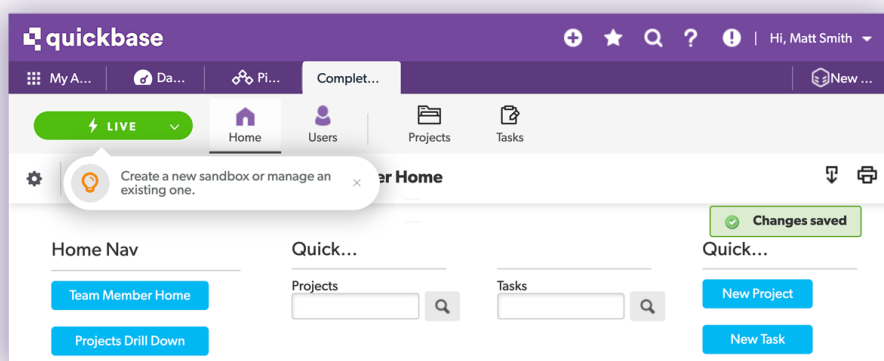
Quickbase je low-code platforma pro vývoj aplikací, která byla založena v roce 1999 Joem Ricem, Jimem Salemem a Claude von Roesgenem [9]. Původně byla součástí produktového portfolia společnosti Intuit, ale v roce 2016 byla prodána soukromé investiční společnosti Welsh, Carson, Anderson & Stowe a později v roce 2019 převzata společností Vista Equity Partners.

Platforma Quickbase je založena na cloudu, který umožňuje uživatelům vytvářet vlastní aplikace a automatizovat obchodní procesy bez rozsáhlých znalostí programování. Platforma poskytuje vizuální rozhraní, řadu předdefinovaných šablon a nástrojů, které umožňují uživatelům vytvářet aplikace, které slouží například k ří-

zení pracovních postupů, zefektivnění podnikových operací, správě objektů a řešení vztahů se zákazníkem.

Uživatelské rozhraní je navrženo pro uživatele bez větších technických znalostí, aby se snadno vyznali v prostředí a mohli bez problému vytvářet a spravovat své aplikace. Při tvorbě designu uživatelského rozhraní aplikace mohou uživatelé využít výběr z mnoha šablon nebo vytvořit si svoji aplikaci úplně od začátku. Díky funkci „drag and drop“ uživatelé lehce vybírají ze sady komponent, které následně jen přetáhnou do své aplikace. Quickbase umožňuje vytváření reportů, grafů, aplikování filtrů a agregací a dashboardů pro vizualizaci a přehlednost dat v reálném čase.

Quickbase zahrnuje funkce pro efektivní plánování, sledování a spolupráci na projektech, což umožňuje uživatelům vytvářet plány projektů, definovat úkoly a stanovit milníky. Další důležitou funkcí je možnost definování pravidel a spouštěčů pro automatické provádění akcí na základě určitých podmínek, jako jsou upozornění, aktualizace záznamů, generování zpráv a zahájení úloh nebo schvalovacích procesů. Uživatelé mohou definovat tato pravidla a spouštěče pomocí nástrojů poskytovaných platformou, které jsou založeny na principu „vyber a klikni“ (point and click). Platforma rovněž podporuje spolupráci mezi členy týmu tím, že umožňuje více uživatelům současně přistupovat k datům a aktualizovat je.



Obrázek 2.8: Uživatelské rozhraní Quickbase

V oblasti správy dat platforma využívá svou integrovanou databázi, kde jsou data reprezentována v tabulkách, a uživatelé mohou vytvářet různé typy vztahů mezi tabulkami a poli. Z hlediska ochrany dat Quickbase provádí pravidelné bezpečnostní audity a implementuje šifrování dat v databázi. Dále platforma nabízí

nástroje na nastavení uživatelských rolí a oprávnění, které jasně definují, kdo může číst, vkládat a upravovat data.

Quickbase dovoluje rozsáhlé možnosti pro integraci s externími systémy a aplikacemi. Uživatelé mohou využívat předdefinované integrační nástroje pro jednoduché propojení, stejně jako pokročilé RESTful API pro vývoj specifických integrací.

Quickbase je navržen pro efektivní škálování, což zahrnuje jak horizontální škálování platformy, tak vertikální škálování aplikací. Platforma pravidelně aktualizuje fyzický hardware, aby zajistila rychlý a hladký běh aplikací. Současně dochází k vylepšením runtime engine, díky čemuž nové i stávající aplikace automaticky získávají vyšší výkon. Aplikace jsou navíc optimalizovány pro zefektivnění procesů, jako je eliminace nevyužívaných dat, čištění nepotřebných polí a optimalizace uživatelského rozhraní. Tyto kroky minimalizují zbytečné výpočty a zvyšují celkovou efektivitu aplikací na platformě Quickbase.

Uživatelé platformy mají přístup k podpoře v podobě návodů, vzdělávacích materiálů, technické dokumentace a video tutoriálů. Pro osobnější přístup je uživatelům poskytnuta pomoc prostřednictvím online chatu nebo telefonické podpory. Pro rozvoj komunity a sdílení informací jsou k dispozici online fóra a webináře.

Aplikace vytvořené na platformě Quickbase jsou obvykle sdíleny v rámci podniku, kde jsou přístupné uživatelům, kterým bylo přiděleno přístupové právo.

V nabídce Quickbase se nachází pro nové uživatele zkušební verze na 30 dní zdarma, dále je v nabídce dva plány s pevnou začáteční cenou plán Team a plán Business. Poslední plán Enterprise je sestaven zcela na konkrétních požadavcích zákazníka a podle toho je i stanovena cena.

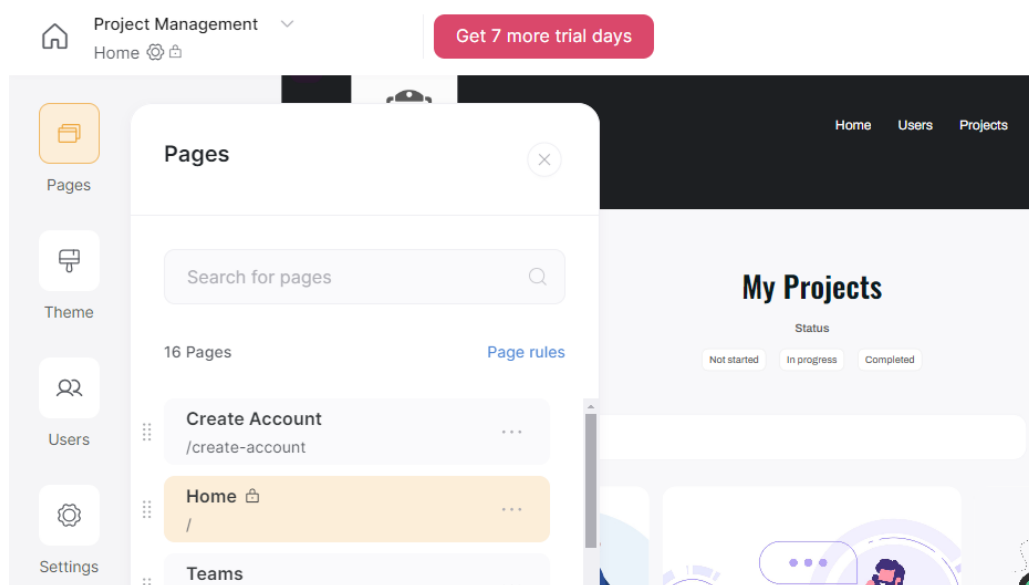
2.3.9 Softr

Web	www.softr.io
Licence	Freemium, Basic 58 USD za měsíc, Professional 167 USD za měsíc, Business 323 USD za měsíc, Enterprise s nastavitelnou cenou
Datum poslední verze	1.12. 2023
Výrobce	Softr Platforms GmbH
Typ vstupu	data z tabulkových editorů, vizuální modelování, integrace databáze, logické a podmíněné vstupy, API a externí integrace, nastavování vlastností, programový kód
Výstupní technologie	Webové aplikace, PWA
Cílové platformy	Web, iOS, Android

Tabulka 2.9: Informace o platformě Softr

Softr je platforma, která umožňuje uživatelům přeměnit data z tabulkových procesorů, jako je Airtable nebo Google Sheets, na klientské portály a interní nástroje bez nutnosti znalostí kódování [10]. Uživatelé mohou s pomocí předdefinovaných bloků snadno vytvářet seznamy, grafy, formuláře, tabulky, kalendáře a mapy. Softr podporuje také vytváření aplikací s přizpůsobitelnými oprávněními a úrovněmi přístupu, které jsou zabezpečené a snadno publikovatelné na různých zařízeních.

Platforma nabízí jednoduché a intuitivní uživatelské rozhraní, díky kterému je proces vytváření a designu UI/UX přístupný i pro nevyvojáře. Vývoj aplikací je zjednodušen díky možnosti výběru ze stovek předdefinovaných šablon nebo možnosti začít od nuly.



Obrázek 2.9: Uživatelské rozhraní Softr

Co se týče funkcionalit a komponent, Softr poskytuje rozmanitý výběr statických i dynamických bloků, včetně pokročilých formulářů s podmíněnou logikou, nástrojů pro vizualizaci dat, jako jsou grafy a diagramy, a prvků pro interakci uživatelů, jako jsou komentáře a platební systémy. Zabezpečení aplikace je podpořeno možností konfigurace oprávnění pro editaci dat a v blízké budoucnosti je plánována funkce bezpečnostního auditu.

Integrace s externími službami a API je v Softr usnadněna díky možnosti propojení s různými nástroji a službami, jako jsou Zapier, Google Analytics a Stripe.

Pokud jde o škálovatelnost a výkon, Softr umožňuje uživatelům stavět a publikovat neomezený počet aplikací a spravovat velký počet záznamů v dynamických blocích, ačkoli mohou existovat omezení v závislosti na použitém tarifu.

Podpora a komunita Softr jsou silnými stránkami této platformy, které zahrnují kvalitní dokumentaci, živou komunitu uživatelů a možnost přímého spojení s týmem Softr prostřednictvím komunitního fóra a živého chatu. Uživatelé mají také možnost účastnit se workshopů a vzdělávacích akcí a získat přístup k personalizované technické podpoře.

Aplikace vytvořené na Softr lze publikovat na různých zařízeních a je možné je stáhnout jako progresivní webové aplikace (PWA).

Model cenové politiky Softr zahrnuje různé cenové plány v závislosti na potřebách a velikosti týmu uživatele, přičemž nabízí možnosti od základních až po pokročilé tarify, které obsahují další funkce a podporu.

2.3.10 Zoho Creator

Web	www.zoho.com
Licence	15denní zkušební verze zdarma, Standard 8 USD za měsíc, Professional 20 USD na uživatele za měsíc, Enterprise 25 USD na uživatele za měsíc, Flex s nastavitelnou cenou
Datum poslední verze	15.12 2023
Výrobce	Zoho Corporation
Typ vstupu	data z tabulkových editorů, vizuální modelování, integrace databáze, logické a podmíněné vstupy, API a externí integrace, nastavování vlastností, programový kód
Výstupní technologie	Webové aplikace, Mobilní aplikace, PWA
Cílové platformy	Web, iOS, Android

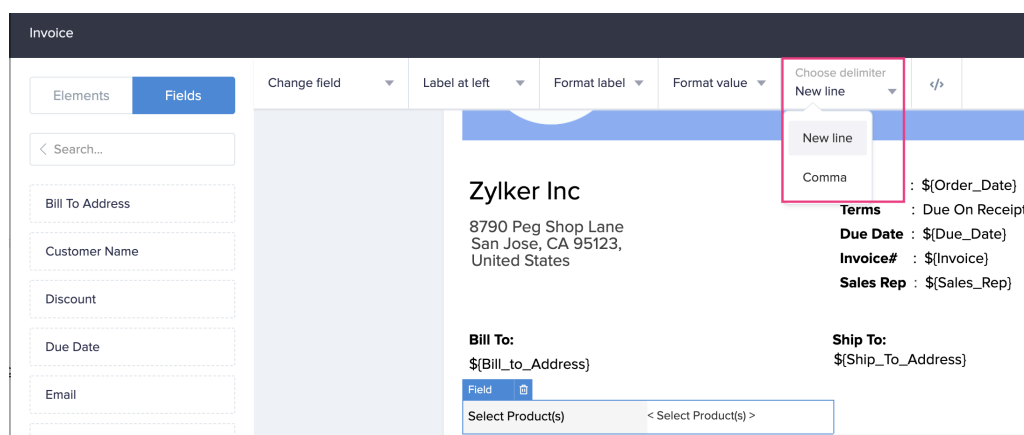
Tabulka 2.10: Informace o platformě Zoho Creator

Zoho Creator je low-code platforma, která byla založena společností Zoho Corporation. Tato platforma poskytuje své vývojové prostředí pro vytváření a správu podnikových aplikací uživatelům s nízkými nebo žádnými technickými znalostmi [11]. Zoho Creator eliminuje složitost spojenou s celým životním cyklem vývoje aplikace, což napomáhá k tomu, že vytváření aplikace je rychlejší a může se zúčastnit kdokoli.

Uživatelské rozhraní využívá nízkokódového přístupu, což umožňuje uživatelům snadno a rychle vytvářet aplikace pomocí grafického rozhraní. Uživatelé mohou vizuálně uspořádat prvky, jako jsou tlačítka, vyhledávače, tabulky a další interaktivní komponenty. Do začátku mohou uživatelé využít rozsáhlou sbírku šablon, kteří se snadno instalují a přizpůsobují potřebám uživatele. Proces vytváření designu aplikace dále zahrnuje vizuální stavbu struktury aplikace a datových toků, kde uživatelé mohou ukládat různá data a vytvářet vlastní datové modely. Aplikace

vytvořené na Zoho Creator automaticky podporují multi-device prostředí, které umožňuje fungování aplikace na různých zařízeních.

Automatizace procesů v Zoho Creator je klíčovou funkcí, díky níž mohou uživatelé vytvářet pracovní postupy a automatizovat opakující se úkoly. Automatizované pracovní postupy mohou reagovat na konkrétní události nebo podmínky v aplikaci. Například, aplikace může automaticky aktualizovat záznamy v CRM, odesílat upozornění nebo přiřazovat úkoly.



Obrázek 2.10: Uživatelské rozhraní Zoho Creator

Data aplikace jsou v Zoho Creator ukládána do vlastního databázového systému platformy. Platforma zde umožňuje uživatelům sbírat, spravovat a analyzovat data. Bezpečnost zajišťuje zálohování dat, ověřování a správu přístupu, včetně role-based přístupových práv.

Platforma poskytuje více než 550 předdefinovaných integrací, což umožňuje uživatelům propojit nástroje a služby, které již používají, a provozovat své podnikání z jedné platformy. Uživatelé mohou využívat vestavěné konektory nebo vytvářet vlastní integrace pomocí rozhraní API.

Aplikace vytvořené na Zoho Creator jsou bezproblémově škálovatelné podle rostoucích podnikových požadavků. Aktuálně je na Zoho Creator vytvořeno přes 7 milionů aplikací, které používá více než 14 000 zákazníků ve více než 180 zemích, což svědčí o síle této platformy.

Uživatelská podpora platformy nabízí svým uživatelům online školení, vzdělávací materiály, technickou dokumentaci, návody a video tutoriály. K dispozici je také podpora prostřednictvím e-mailu, telefonu a live chatu, což uživatelům dává rychlou možnost získat pomoc při potížích nebo dotazech. Další výraznou podporou pro uživatele může být aktivní komunita Zoho Creator, která klade otázky, sdílí nápady a pomáhá s případnými dotazy.

Publikování aplikací je jednoduché, stačí si vybrat prostředí, ve kterém aplikaci nasadit, a následně publikovat. Aplikace vytvořené jako webové jsou automaticky dostupné jako nativní mobilní aplikace pro Android a iOS, které lze publikovat prostřednictvím Google Play nebo App Store.

Platforma nabízí tři hlavní plány Standard, Professional a Enterprise, každý s různými funkcemi a limity a pevně nastavenou cenou. K dispozici je také Flex plán pro uživatele s vlastními požadavky. Pro začínající uživatele platforma poskytuje 15denní bezplatnou zkušební verzi.

2.4 Tabulka nalezených nástrojů

Jméno	Typ	Free modely	Podporované Jazyky	Výstup
Adalo	no-code	Freemium	HTML, CSS, SQL a JavaScript	PWA, Webové a Mobilní aplikace
Bubble	no-code	Freemium	HTML, CSS, SQL, Python a JavaScript	PWA a Webové aplikace
Glide	no-code	Freemium	HTML, CSS, SQL a JavaScript	PWA, Webové a Mobilní aplikace
Google AppSheet	no-code	Freemium	AppSheet Expression Language, HTML, CSS, SQL a JavaScript	PWA, Webové a Mobilní aplikace
Mendix	low-code	Freemium	HTML, CSS, Java, JavaScript, SQL, React	PWA, Webové a Mobilní aplikace
Microsoft Power Apps	low-code	Freemium	HTML, CSS, JavaScript, SQL, Power Fx	Mobilní aplikace
OutSystems	low-code	Freemium	HTML, CSS, Java, JavaScript, OSL, SQL a C#	PWA, Webové a Mobilní aplikace
Softr	no-code	Freemium	HTML, CSS a JavaScript	PWA, Webové a Mobilní aplikace
Quick Base	low-code	Trial	HTML, CSS, JavaScript, Python, C#, Golang a SQL	PWA, Webové a Mobilní aplikace
Zoho Creator	low-code	Trial	Deluge, HTML, CSS, Java, JavaScript, SQL	PWA, Webové a Mobilní aplikace

Tabulka 2.11: Tabulka nalezených platforem

2.5 Umělá inteligence

Umělá inteligence (AI) je pokročilá oblast počítačové vědy, která se zaměřuje na vytváření systémů schopných vykonávat úlohy, jež obvykle vyžadují lidskou inteligenci. Tato technologie umožňuje počítačům a strojům učit se z dat, rozpoznávat vzory, dělat rozhodnutí a řešit problémy s minimálním lidským zásahem. Klíčovými aspekty AI jsou počítačové vidění, robotika a strojové učení, kde jsou zahrnuty klíčové techniky jako hluboké učení, neuronové sítě a zpracování přirozeného jazyka (NLP). Tyto technologie umožňují aplikacím, jako jsou virtuální asistenti, autonomní vozidla a automatizované rozhodovací systémy, efektivněji a snadněji interagovat s uživateli a okolním světem.

Využití umělé inteligence se neustále rozšiřuje, transformuje průmyslové odvětví, zlepšuje efektivitu operací a otevírá nové možnosti pro inovace. Například v oblasti vývoje softwaru nástroje založené na AI, jako je GitHub Copilot nebo ChatGPT od OpenAI, umožňují vývojářům generovat kódové fragmenty a celé programové struktury na základě přirozeného jazyka. To znamená, že vývojáři mohou specifikovat, co chtějí, aby výsledný kód dělal, ve formě komentářů nebo jednoduchých instrukcí, a nástroj vygeneruje odpovídající kód. Tím se výrazně snižuje čas strávený psaním a laděním kódu, což umožňuje vývojářům soustředit se na strategičtější aspekty projektů. Jak ukázala studie prováděná vědeckými pracovníky z Microsoft Research [12], nástroje pro asistenci programování založené na velkých jazykových modelech, jako je GitHub Copilot, mohou významně zrychlit proces vývoje softwaru tím, že automatizují opakující se úlohy, generují kompletní funkční těla a testovací případy. Uživatelské zkušenosti zaznamenané v této studii dokládají, že vývojáři obecně ocenili schopnosti těchto nástrojů a shledali je jako pozitivní přínos k vývojovému procesu, přestože nebyly zaznamenány silné efekty na čas dokončení úkolů nebo správnost kódu.

V následujících podkapitolách jsou popsány vybrané umělé inteligence, které se využívají pro generování kódu při vývoji softwaru.

2.5.1 ChatGPT

ChatGPT je pokročilý jazykový model vyvinutý společností OpenAI, založený na architektuře Generative Pre-trained Transformer (GPT). Je navržen tak, aby generoval koherentní a kontextuálně relevantní texty na základě zadaných pokynů nebo dotazů v přirozeném jazyce. Jeho schopnost generovat dobře vypadající text k široké škále témat a formulovat odpovědi, které napodobují lidský způsob vyjadřování, umožňuje jeho využití v mnoha aplikacích, včetně chatbotů, automatického generování obsahu a asistování ve vývoji softwaru. Díky obrovskému množství dat, na kterých byl trénován, má ChatGPT rozsáhlé znalosti o programování, což zahr-

nuje různé programovací jazyky, vývojové praktiky a algoritmické koncepty, čímž se stává cenným nástrojem pro programátory a vývojáře.

V oblasti vývoje softwaru ChatGPT pomáhá generováním kódu, laděním, vysvětlováním kódových fragmentů a poskytováním návodů nebo dokumentace [13]. Uživatelé mohou modelu zadat dotazy nebo popisy požadované funkcionality v přirozeném jazyce, na které ChatGPT odpovídá generováním příslušných kódových fragmentů nebo podrobných vysvětlení. Tato schopnost snižuje čas potřebný pro vývoj a testování kódu, umožňuje rychlé prototypování a pomáhá vývojářům lépe porozumět a zvládnout jejich programovací problémy [13]. Přestože kód generovaný ChatGPT může vyžadovat další revizi a validaci, aby se zajistila jeho přesnost a bezpečnost, ChatGPT výrazně zvyšuje produktivitu a efektivitu vývojářů tím, že jim poskytuje okamžitou zpětnou vazbu a podporu během celého vývojového procesu, což potvrdili studenti ve výzkumu o názoru studentů na použití ChatGPT ve výuce programování [14].

2.5.2 GitHub Copilot

GitHub Copilot, vyvinutý GitHubem ve spolupráci s OpenAI, představuje průlomový nástroj pro generování kódu, který využívá AI model Codex k převodu instrukcí v přirozeném jazyce na programový kód. Tento nástroj, integrovaný přímo do vývojových prostředí, poskytuje vývojářům návrhy kódu v reálném čase, čímž může výrazně zvýšit efektivitu vývoje softwaru. Díky podpoře širokého spektra programovacích jazyků a frameworků je Copilot neocenitelným pomocníkem pro rychlejší a plynulejší vývojový proces, umožňující programátorům soustředit se na složitější aspekty jejich projektů. Detailnějším porovnáním GitHub Copilota s běžným programováním se zabývá i konferenční příspěvek *The Recent Trends of Research on GitHub Copilot: A Systematic Review* [15], který zjistil, že zatímco Copilot zvýšil produktivitu, kvalita generovaného kódu byla horší. Dále odhalil, že Copilot poskytl programátorům užitečný výchozí bod, ale byla potřeba pomoc s pochopením, úpravou a laděním generovaných úryvků kódu. Celkovým závěrem je významný příslib při generování užitečného kódu s vysokou mírou úspěšnosti a zvýšení produktivity uživatelů. Je však třeba řešit případné nedostatky, jako je kvalita a přehlednost generovaného kódu, které by bylo možné dále zjednodušit.

GitHub Copilot poskytuje cenou podporu při generování kódu, ale jeho účinné využití vyžaduje základní programátorské znalosti. Uživatelé bez hlubších znalostí programování mohou narazit na výzvy při interpretaci a integraci vygenerovaného kódu do projektů. Pro efektivní využití Copilota je důležité mít alespoň základní porozumění programovacím principům, což umožňuje ověřit správnost, efektivitu a bezpečnost kódu. Tím pádem, i když Copilot může usnadnit některé aspekty vývoje, nenahrazuje potřebu základního vzdělání a praxe v softwarovém inženýrství.

2.5.3 Zhodnocení AI jako nástroje na tvorbu aplikací

Přístup k umělé inteligenci jako nástroji pro tvorbu aplikací představuje významný krok směrem k inovacím a efektivitě ve vývoji softwaru. Nástroje jako ChatGPT a GitHub Copilot demonstrují, jak AI může automatizovat generování kódu, poskytovat podporu při ladění a vysvětlování kódových struktur a tím usnadnit rychlejší prototypování a vývoj. Tyto technologie nabízejí slibné možnosti pro zjednodušení některých aspektů programování a umožňují vývojářům soustředit se na strategičtější a kreativní části procesu vývoje.

Nicméně, při zhodnocení AI jako nástroje pro tvorbu aplikací je důležité uznat její omezení, zejména v kontextu zapojení lidí bez programátorských znalostí. Ačkoli AI umožňuje určitou míru automatizace, hlubší porozumění programovacím jazykům, logice a architektuře softwaru je nezbytné pro efektivní využití vygenerovaného kódu. Bez těchto základních znalostí se uživatelé mohou potýkat s výzvami při integraci a validaci kódu, což omezuje jejich schopnost účinně přispívat k vývoji aplikací. To znamená, že i když AI slibuje zpřístupnění vývoje softwaru širšímu spektru lidí, plná realizace tohoto potenciálu vyžaduje, aby uživatelé měli alespoň základní technické dovednosti a porozumění programování.

V závěru, AI představuje cenný nástroj, který může výrazně podpořit proces tvorby aplikací, ale nejedná se o univerzální řešení pro všechny aspekty vývoje softwaru. Pro plné využití možností, které AI nabízí, je stále nezbytné, aby vývojáři a ti, kdo se chtějí podílet na tvorbě aplikací, investovali do získávání a rozvíjení svých programátorských dovedností. Takto AI slouží spíše jako doplňkový nástroj, který zvyšuje efektivitu a inovaci, než jako kompletní náhrada lidského vývojáře.

UX nástrojů pro tvorbu aplikací

3

V současné době, kdy digitální technologie pronikají do všech aspektů našeho života, se stává automatizovaná tvorba aplikací nezbytným nástrojem pro zefektivnění a zjednodušení vývojového procesu. Nástroje pro tvorbu aplikací s minimální potřebou psaní programového kódu umožňují širokému spektru uživatelů, od profesionálů po amatéry, rychle vytvářet a iterovat softwarové projekty. V tomto kontextu se UX (User Experience) stává klíčovým faktorem, který ovlivňuje nejen spokojenost a produktivitu uživatelů, ale také úspěšnost a přijetí vytvořených aplikací.

Přestože je význam UX dobře znám v oblasti grafického rozhraní aplikací, je třeba důkladněji prozkoumat jeho roli v nástrojích pro automatizovanou tvorbu aplikací. UX v těchto nástrojích se netýká pouze intuitivního designu a snadnosti použití, ale také zahrnuje aspekty, jako je přístupnost, personalizace a schopnost nástroje podporovat uživatele v procesu vývoje od konceptu po finální produkt. Vzhledem k tomu, že tato oblast stále nabízí nové možnosti a výzvy, je průzkum existující literatury o UX v nástrojích pro tvorbu aplikací klíčovým krokem k pochopení současného stavu a k identifikaci oblastí potřebujících další výzkum.

Tato kapitola se zaměřuje na analýzu a syntézu existující literatury v oblasti UX. Cílem je nejen poskytnout přehled stávajících poznatků a trendů, ale také identifikovat klíčové faktory úspěchu a oblasti, které vyžadují další pozornost. Přezkoumáním širokého spektra zdrojů, od akademických studií po průmyslové zprávy, se snažíme nabídnout komplexní pohled na to, jak UX ovlivňuje vývoj aplikací pomocí těchto nástrojů a jak může být UX vylepšen pro maximální efektivitu a uživatelskou spokojenost.

3.1 Teorie UX

User Experience (UX) je široký pojem, který se vztahuje na všechny aspekty interakce uživatele s produktem, službou nebo systémem, včetně jakéhokoli softwaru, webových stránek nebo aplikací [16]. Základní definice UX zahrnuje nejen to, co uživatelé vidí, ale také to, jak se cítí během používání produktu, jak snadno mohou

dosáhnout svých cílů a jak efektivní je celkový proces. Cílem UX designu je tedy vytvořit produkty, které jsou nejen použitelné, přístupné a užitečné, ale také poskytují pozitivní zážitky, které uživatele poutají a podporují. Toto zahrnuje důkladné pochopení potřeb uživatelů, kontextu použití a omezení, s nimiž se mohou setkat, stejně jako integraci těchto poznatků do designového a vývojového procesu [17].

Principy UX designu jsou klíčové směrnice, které pomáhají designérům a vývojářům vytvářet efektivnější a přívětivější uživatelské rozhraní. Mezi základní principy patří uživatelský výzkum pro hlubší pochopení cílového publika, iterativní design umožňující neustálé testování a vylepšování, a přístupnost, zajišťující, že produkty jsou dostupné a použitelné pro co nejširší spektrum uživatelů. Dále je důležitá jednoduchost a intuitivnost, což znamená odstranění nepotřebných prvků, které by mohly uživatele zmást nebo odradit od používání. Nakonec, konzistence a předvídatelnost v designu usnadňují učení se a používání produktu, což vede k lepší celkové uživatelské zkušenosti. Tyto principy společně tvoří základ, který umožňuje tvorbu produktů nejen vyhovujících potřebám uživatelů, ale zároveň jim přinášející pozitivní zážitky z jejich používání.

3.2 Modely a metody hodnocení UX

Modely a metody hodnocení UX jsou základními nástroji, které designéři a výzkumníci používají k zajištění, že produkty splňují potřeby a očekávání jejich uživatelů. Jedním z nejznámějších přístupů k hodnocení použitelnosti je Jakob Nielsenův seznam deseti heuristik pro použitelnost, který poskytuje základní principy pro návrh uživatelského rozhraní [18]. Dalším důležitým modelem je UX Honeycomb od Petera Morvilla, který se nezaměřuje pouze na použitelnost ale rozšiřuje perspektivu UX o další složky, jako jsou užitečnost, žádoucnost, přístupnost, věrohodnost, naleznutelnost a hodnota [19].

Co se týče metod hodnocení, existuje široké spektrum technik sahající od kvantitativních přístupů, jako jsou dotazníky a analýzy použitelnosti, po kvalitativní metody, jako jsou rozhovory, pozorování a testování použitelnosti s uživateli. Tyto metody pomáhají odhalit, jak uživatelé interagují s produktem a jaké mají při této interakci zážitky, což umožňuje identifikovat oblasti pro zlepšení.

Při výběru konkrétní metody hodnocení je důležité zvážit cíle výzkumu, charakter produktu a dostupné zdroje. Například, A/B testování a sledování pohybu očí poskytují přesné kvantitativní údaje o tom, jak uživatelé interagují s různými aspekty designu, zatímco metody jako jsou persona a scénáře umožňují hlubší porozumění motivacím a chování uživatelů [20]. Využití kombinace těchto metod může poskytnout komplexní pohled na UX a přispět k vytvoření produktů, které jsou nejen technicky funkční, ale také poskytují pozitivní a smysluplné zážitky pro uživatele.

3.2.1 Nielsenovo deset heuristik

Jakob Nielsen, uznávaný expert na použitelnost, formuloval deset heuristik pro návrh uživatelského rozhraní, které slouží jako základní principy pro zlepšení UX designu. Tyto heuristiky, i přestože byly původně navrženy v 90. letech, zůstávají i dnes klíčovými směrnicemi pro evaluaci a návrh interaktivních systémů [18].

1. Viditelnost stavu systému

System by měl vždy informovat uživatele o aktuálním stavu prostřednictvím vhodné zpětné vazby v rozumném časovém rámci.

2. Shoda mezi systémem a reálným světem

System by měl mluvit jazykem uživatele, používat slova, fráze a koncepty, které jsou uživateli známé, nikoli interní technický jargon.

3. Uživatelská kontrola a svoboda

Uživatelé často potřebují možnost odvolat nebo opustit nesprávně zvolenou operaci; systém by měl nabízet jednoduchý způsob, jak to provést bez rozsáhlých dialogů.

4. Konzistence a standardy

Uživatelé by neměli být nuceni přemýšlet, zda různá slova, situace nebo akce znamenají to samé. Postupujte podle platformově konzistentních konvencí.

5. Prevence chyb

Lepší než dobré zprávy o chybách je pečlivě navržený systém, který předchází problémovým situacím.

6. Rozpoznání je lepší než připomínání

System by měl dělat viditelné objekty, akce a možnosti, uživatel by neměl muset pamatovat informace z jedné části dialogu do druhé.

7. Flexibilita a efektivita použití

Systemy by měly být flexibilní a umožňovat uživatelům přizpůsobit časté akce, aby efektivněji dosáhli svých cílů.

8. Estetický a minimalistický design

Dialogy by neměly obsahovat informace, které jsou irelevantní nebo zřídka potřebné. Každý dodatečný prvek v uživatelském rozhraní může rozptylovat a zbytečně komplikovat.

9. Pomoc uživatelům pochopit, diagnostikovat a řešit chyby

Zprávy o chybách by měly být vyjádřeny jasným jazykem, popsat problém a navrhnout konstruktivní řešení.

10. Náповěda a dokumentace

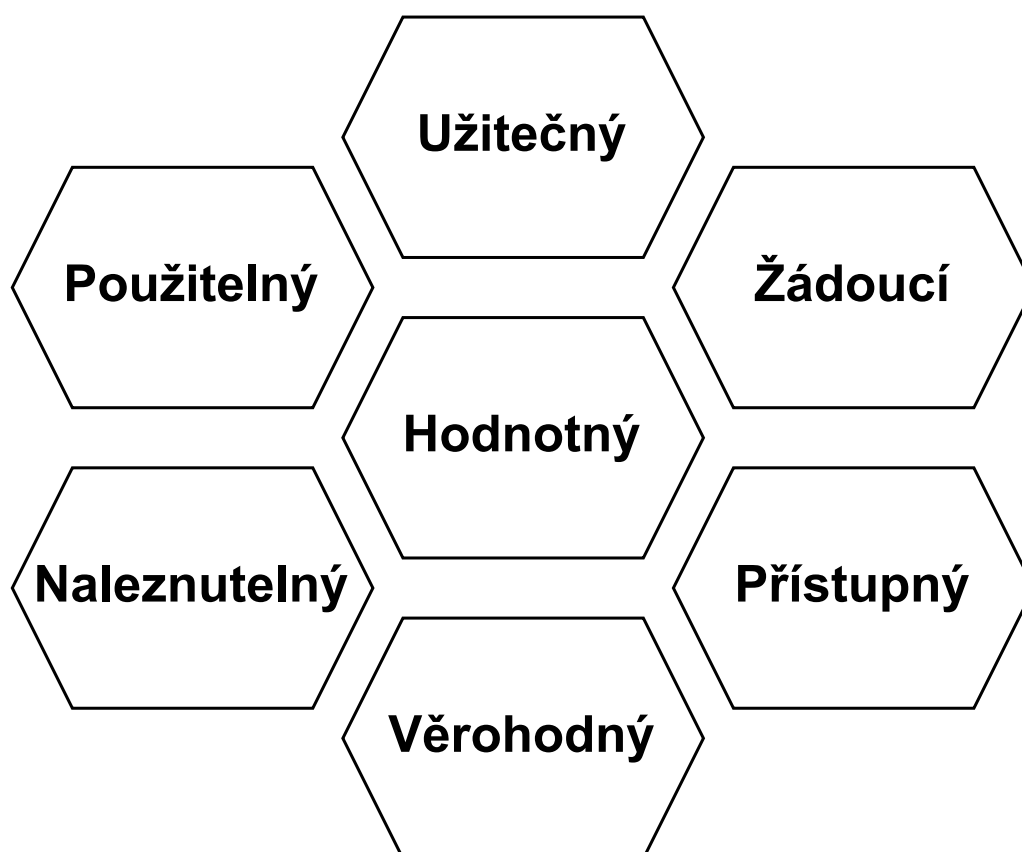
I když je lepší, když systém může být použit bez jakékoli nápovědy, měla by být dostupná kompletní nápověda a dokumentace, která je snadno vyhledatelná, zaměřená na úkol uživatele, a která obsahuje kroky k řešení běžných problémů.

Tyto heuristiky společně poskytují komplexní rámec pro evaluaci a zlepšování uživatelského rozhraní, což vede k vytvoření intuitivnějších, efektivnějších a celkově příjemnějších digitálních produktů [21].

3.2.2 UX Honeycomb od Petera Morvilla

Peter Morville, jeden z průkopníků v oblasti architektury informací, vytvořil model UX známý jako „Honeycomb“ (plástev), který rozšiřuje pohled na uživatelskou zkušenost oproti tradičním přístupům zaměřeným výhradně na použitelnost. Tento model se skládá ze sedmi základních aspektů, které by měly být zváženy při návrhu digitálních produktů a služeb. Seskupení těchto aspektů do tzv. plástve je vidět na obrázku 3.1. Podle modelu by služba nebo produkt měly být užitečný, použitelný, žádoucí, přístupný, věrohodný, naleznutelný a hodnotný [19]. Morville tyto kvality rozložil do formy plástve s centrálním šestiúhelníkem obklopeným dalšími šesti, což nám umožňuje zkoumat, jak tyto kvality vzájemně interagují. Například, pokud nástroj nabízí minimalistický design a jednoduchost použití, může přitahovat širší spektrum uživatelů a usnadnit více lidem práci s nástrojem, což ukazuje rostoucí vztah mezi žádoucností a přístupností.

Model Honeycomb slouží designérům a vývojářům jako důležitý nástroj, který jim pomáhá vytvářet produkty a služby, které jsou technicky funkční, ale také poskytují uživatelům smysluplné a pozitivní zážitky. Díky tomuto modelu mohou tvůrci lépe porozumět tomu, co uživatelé očekávají od digitálních produktů, a jak tato očekávání splnit skrze komplexní přístup k designu, který zahrnuje široké spektrum faktorů ovlivňujících uživatelskou zkušenost. Morvillův přístup zdůrazňuje, že úspěšný digitální produkt by měl být nejen snadno použitelný, ale také by měl uživatelům poskytovat hodnotu, být pro ně přitažlivý a snadno dostupný, což v konečném důsledku vede k větší spokojenosti a loajalitě uživatelů.



Obrázek 3.1: UX Honeycomb od Petera Morvilla

3.3 UX v nástrojích pro tvorbu aplikací

UX v nástrojích pro tvorbu aplikací hraje zásadní roli při zpřístupnění vývoje softwaru pro širší spektrum uživatelů, kteří nemají zkušenosti s vývojem softwaru. Tyto nástroje, které často využívají vizuální programovací jazyky nebo „drag and drop“ rozhraní, se musí zaměřit na poskytování intuitivního, přístupného a flexibilního uživatelského zážitku, aby maximalizovaly produktivitu uživatelů a minimalizovaly frustraci z učení se novým systémům. Kvalitní UX v těchto nástrojích podporuje rychlé prototypování a iteraci, což je zásadní pro agilní vývojové cykly, a zajišťuje, že uživatelé mohou snadno převádět své nápady do fungujících aplikací bez psaní komplexního kódu. To nejen zvyšuje přístupnost vývoje softwaru pro nevyvojáře, ale také umožňuje zkušeným vývojářům rychleji testovat a implementovat nové koncepty.

Z hlediska UX designu je výzvou pro tvůrce těchto nástrojů nalézt rovnováhu mezi množstvím funkcionalit a jednoduchostí. Nástroje pro tvorbu aplikací by měly

být dostatečně robustní, aby podporovaly složité vývojové požadavky, zároveň však musí zůstat dostatečně jednoduché pro uživatele, kteří nemají technické znalosti. To vyžaduje pečlivé zvážení uživatelských rozhraní, dokumentace a vzdělávacích zdrojů, stejně jako vývoj intuitivních systémů nápovědy a podpory, které mohou uživatele provést procesem vývoje. Efektivní implementace UX v těchto nástrojích také zahrnuje průběžné zpětné vazby od uživatelů a jejich zapojení do vývojového procesu, což umožňuje neustálé vylepšování a adaptaci nástrojů tak, aby lépe vyhovovaly potřebám uživatelů [20]. Tímto způsobem UX design nejenže zlepšuje jednotlivé nástroje, ale také přispívá k celkovému posunu paradigmatu ve způsobu, jakým přistupujeme k vývoji softwaru, směrem k větší inkluzivitě a uživatelské orientaci.

Srovnání nalezených nástrojů

4

Tato kapitola se věnuje podrobnému srovnání no-code a low-code nástrojů, které byly identifikovány jako klíčové platformy na trhu pro rychlý a efektivní vývoj softwarových aplikací s minimální potřebou psaní kódu. Cílem je poskytnout ucelené srovnání nástrojů, zahrnující informace o dostupnosti, cenové politice, klíčových funkcionalitách a dalších aspektech, které jsou zásadní při výběru vhodné platformy pro vývoj aplikací.

V úvodu je klíčové zdůraznit, že srovnání nástrojů je zvlášť organizováno pro no-code a low-code platformy. Toto rozdělení je podstatné pro rozlišení mezi dvěma základními přístupy k vývoji aplikací. No-code platformy jsou zaměřeny primárně na uživatele bez programátorských dovedností a umožňují tvorbu aplikací výhradně prostřednictvím vizuálního rozhraní, zatímco low-code platformy jsou navrženy s ohledem na vývojáře a analytiku, nabízející větší flexibilitu a možnosti integrace, avšak za cenu mírně vyšší složitosti.

Na konci této kapitoly budou vybrány tři platformy pro hlubší analýzu, kde se budou platformy hodnotit podle použitelnosti a UX vybraných nástrojů na základě připravené ukázky. Tento výběr je motivován snahou poskytnout čtenářům detailní přehled.

4.1 Srovnání no-code platformem

V následujících podkapitolách podrobně srovnáme dostupnost a cenu no-code platformem, stejně jako funkcionality nabízené v jejich bezplatných verzích. Cílem je poskytnout uživatelům jasný přehled o tom, jaké možnosti tyto platformy nabízejí a jak mohou nejlépe vyhovovat jejich specifickým potřebám.

4.1.1 Srovnání podle dostupnosti a ceny

Adalo, Glide a Softr se vyznačují svým příjemným prostředím a dostupností i pro uživatele bez programovacích schopností. Adalo a Glide nabízí integrované řešení

Jméno	Bezplatné modely	Placené modely
Adalo	Freemium	od 45 do 250 USD za měsíc
Bubble	Freemium	od 32 do 399 USD za měsíc
Glide	Freemium	od 60 do 1200 USD za měsíc
Google AppSheet	Freemium	od 5 do 10 USD za měsíc
Softr	Freemium	od 58 do 323 USD za měsíc

Tabulka 4.1: Srovnávací tabulka no-code platformem podle ceny a dostupnosti

databází, což umožňuje uživatelům spravovat data přímo v rámci těchto nástrojů. Na rozdíl od nich, platforma Softr neobsahuje vlastní databázové řešení, ale místo toho umožňuje připojení k externím databázím, jako jsou Airtable nebo Google Sheets.

Bubble se orientuje na tvorbu složitějších webových aplikací, nabízí větší flexibilitu a možnosti pro vývojáře, což může být náročnější pro začátečníky, ale ideální pro tvorbu komplexních projektů bez kódování. Na druhé straně, Google AppSheet integruje silnou podporu pro automatizaci a aplikace založené na datech přímo z ekosystému Google, jako jsou Google Sheets, což je výhodné pro ty, kdo jsou již s těmito nástroji obeznámeni a hledají efektivní způsob, jak digitalizovat své procesy.

Adalo, Bubble, Glide, Google AppSheet a Softr představují různé přístupy k cenění vývoje aplikací bez kódování. Platformy poskytují řadu cenových možností, od freemium modelů až po různé úrovně placených plánů. Adalo nabízí bezplatný přístup s možností upgrade na placené plány, které začínají na 45 USD a sahají až k 250 USD měsíčně [2]. Bubble rovněž poskytuje freemium model a placené modely začínající na 32 USD, které se mohou vyšplhat až na 399 USD měsíčně [3], což ukazuje na široký rozsah funkcí a vysokou úroveň přizpůsobení, které platforma může nabídnout. Poslední cenový model Bubble je určen pro velké podniky, kterým by předdefinované modely nestačily. Cena tohoto modelu se pak odvíjí podle požadovaných funkcí daného podniku. Glide přichází s širokým výběrem cenových plánů, jak pro jednotlivce tak pro týmy, s jedním bezplatným modelem a několika placenými modely s cenami začínajícími v rozmezí od 50 do 1200 USD měsíčně [4]. Google AppSheet, který také nabízí bezplatný přístup k platformě, je zvláště cenově dostupný s cenami mezi 5 a 10 USD měsíčně za uživatele, díky čemuž je atraktivní pro uživatele hledající cenově efektivní řešení integrované do ekosystému Google. A pro náročnější uživatele Google AppSheet poskytuje dva modely s nastavitelnou cenou, která se zvyšuje s počtem požadovaných funkcí [5]. Poslední platforma Softr rovněž poskytuje jako ostatní nástroje svůj bezplatný model. Cenové modely se pak pohybují od 58 USD do 323 USD měsíčně. K tomu platforma nabízí jeden model s nastavitelnou cenou pro velké podniky s velkými potřebami na funkcionalitu [10]. Cenové modely různých platformem jsou navrženy tak, aby vyhovely různým po-

třebám a rozpočtům, poskytující hodnotu jak pro jednotlivé tvůrce, tak pro větší podniky.

4.1.2 Srovnání podle funkcionality bezplatných verzí

Funkce	Adalo	Bubble	Glide	Google AppSheet	Softr
Webové aplikace	ano	ano	ano	ano	ano
Mobilní aplikace	ano	ne	ne	ne	ne
PWA	ano	ne	ano	ne	ne
Testovací aplikace	neomezeno	neomezeno	neomezeno	neomezeno	neomezeno
Sdílení aplikací	neomezeno	neomezeno	3	neomezeno	neomezeno
Publikace aplikací	ne	ne	3	ne	neomezeno
Počet záznamů v databázi	200	200	25000	1000	1000 na tabulku
Integrace	omezené	omezené	omezené	omezené	omezené
Podpora programovacích jazyků	ne	HTML, CSS a JavaScript	ne	AppSheet Expression Language	ne
Propojení s vlastním API	ne	ano	ne	ne	ne
Šablony	ano	ano	ano	ano	ano
Responzivní design	ano	ano	ano	ano	ano

Tabulka 4.2: Srovnání verzí zdarma no-code platforem podle funkcionality

Ve srovnání pětice plánů zdarma no-code nástrojů z hlediska klíčových funkcí a limitů jejich bezplatných verzí je vidět v tabulce 4.2, že umožňují tvorbu webových aplikací, nikoliv však mobilních aplikací. Nabízejí pouze webové aplikace optimalizované pro mobilní zařízení. Kromě nástroje Adalo, který nabízí vytváření i mobilních aplikací a PWA, a nástroje Glide, který umožňuje vytvářet PWA. Všechny nástroje nabízí vytváření neomezeného počtu testovacích aplikací. Co se týče sdílení aplikací, Adalo, Bubble, Softr a Google AppSheet poskytují neomezené možnosti,

zatímco Glide má omezení na tři sdílené aplikace. Nástroj Glide navíc umožňuje publikovat až tři aplikace, zatímco Softr nabízí publikaci aplikací v neomezeném množství, což uživatelům obou platforem poskytuje flexibilitu v závislosti na jejich potřebách. Kapacity databází poskytované různými vývojovými platformami se liší, přičemž Adalo a Bubble umožňují až 200 záznamů na aplikaci [2, 3]. Glide nabízí největší flexibilitu s limitem 25 000 záznamů [4], zatímco Google AppSheet s limitem 1 000 záznamů a Softr rozšiřují možnosti díky tomu, že nemá svojí databázi a pouze připojuje data z externích zdrojů má limit 1 000 záznamů na tabulku [5, 10].

Významné rozdíly mezi platformami lze pozorovat ve schopnosti integrace a programování. Bubble se odlišuje podporou používání programovacích jazyků jako je JavaScript, HTML a CSS. Bubble navíc přidává možnost propojení s vlastními API, zatímco ostatní platformy nemají možnost vlastních API. Dále všechny platformy dávají k dispozici předpřipravené šablony usnadňující rychlé sestavení aplikací a zaručují responzivní design vytvořených aplikací.

Srovnání podle podpory bezplatných verzí

Všechny platformy nabízí podporu uživatelů v podobě komunity a technické dokumentace. Komunity všech nástrojů jsou aktivní a nabízejí uživatelům místo pro diskuse, výměnu zkušeností a řešení problémů, což je zvláště cenné pro uživatele bezplatných verzí těchto platforem. Technické dokumentace obsahují videa a obrázky, aby uživatelům poskytly lepší představu o používání platformy. Glide navíc vyniká díky své interaktivní dokumentaci, která reprezentuje prostředí nástroje pomocí interaktivních prvků, poskytující tak ještě lepší pomoc uživatelům. Na druhé straně, dokumentace Google AppSheet může působit zastarale z hlediska designu, což může uživatelům ztížit orientaci a celkově zhoršit pocit uživatele.

Platformy dále poskytují speciální vzdělávací platformy, které obsahují kurzy, lekce a podrobné průvodce. Adalo má Adalo App Academy, Glide disponuje Glide University, Softr poskytuje Softr Academy a živý chat, a Bubble nabízí Bubble Academy spolu s Bubble Support Center pro komplexnější podporu. Google AppSheet nabízí výuková videa a kurzy, ale ve free verzi je jejich dostupnost omezená, což může limitovat uživatele při pokročilejším vzdělávání v rámci platformy.

4.2 Srovnání low-code platforem

V dalších částech této kapitoly se podrobně zaměříme na srovnání dostupnosti a cenové politiky různých low-code platforem, stejně jako na rozsah funkcí, které nabízejí ve svých bezplatných verzích. Cílem je poskytnout ucelený přehled možností, které tyto nástroje nabízejí, aby budoucí uživatelé mohli lépe pochopit, jak mohou

tyto platformy odpovídají jejich individuálním požadavkům a potřebám v oblasti vývoje aplikací.

4.2.1 Srovnání podle dostupnosti a ceny

Jméno	Bezplatné modely	Placené modely
Mendix	Freemium	od 52,5 do 2 100 EUR za měsíc
Microsoft Power Apps	Freemium	20 USD za měsíc
OutSystems	Freemium	od 32 500 USD za rok
Quickbase	Trial 30 dní	od 35 do 55 USD za měsíc
Zoho Creator	Trial 15 dní	od 8 do 25 EUR za měsíc

Tabulka 4.3: Srovnávací tabulka low-code platforem podle ceny a dostupnosti

Porovnání Mendix, Microsoft Power Apps, OutSystems, Quickbase a Zoho Creator z hlediska cen a dostupnosti odhaluje, jak tyto platformy cílí na různé segmenty trhu s odlišnými potřebami a očekáváními.

Pokud jde o dostupnost, Mendix a OutSystems nabízejí rozsáhlé možnosti pro podnikové uživatele, které mohou vyžadovat složitější zavedení a specifické technické dovednosti, což může znesnadnit přístup. Microsoft Power Apps se vyznačuje dobrou dostupností v rámci firem, které již využívají Microsoft 365, díky snadné integraci a podpoře. Quickbase a Zoho Creator zdůrazňují uživatelskou přívětivost a širokou dostupnost pro různé typy uživatelů, včetně těch bez rozsáhlých technických znalostí, což z nich činí vhodné nástroje pro rychlý vývoj a nasazení aplikací.

Mendix a OutSystems jsou zaměřeny na větší podniky, přičemž jejich cenové modely odrážejí potřebu robustního přizpůsobení a škálovatelnosti. Mendix i OutSystems nabízí jeden freemium model na vyzkoušení aplikace a jeden customizovatelný model pro podniky s velkými požadavky, jehož cena je dána až po výběru všech potřebných parametrů. Mendix dále poskytuje placené modely s cenami dosahujícími od 52,5 EUR až k 2 100 EUR měsíčně [6], což naznačuje široké možnosti využití od základního až po pokročilý vývoj aplikací. OutSystems také poskytuje jeden cenový model ale pouze s ročním předplatným s cenou od 32 500 USD na rok [8]. Microsoft Power Apps poskytuje cenově výhodné řešení pro uživatele, kteří jsou již součástí ekosystému Microsoft, a to s měsíční sazbou 20 USD za uživatele nebo bezplatnou verzí [7]. Quickbase představuje výhodnou volbu pro střední a menší podniky díky transparentním cenám jednotlivých plánů, které začínají na 35 USD až 55 USD měsíčně na uživatele. Quickbase dále poskytuje na vyzkoušení své platformy 30denní trial a model s nastavitelnou cenou s požadavky na míru [9]. Zoho Creator je také atraktivní pro středně velké a malé podniky s cenovými plány v rozmezí od 8 do 25 EUR měsíčně, což podnikům poskytuje snadný vstup do

vývoje aplikací s možností růstu a rozšiřování funkcí podle potřeby, bohužel stejně jako Quickbase nenabízí žádný bezplatný plán, ale jenom 15denní trial a jeden plán s nastavitelnou cenou, který je plně přizpůsobitelný [11].

Tyto platformy tedy nabízejí různé cenové modely, které odpovídají potřebám a finančním možnostem, umožňující jednotlivým uživatelům a podnikům všech velikostí najít řešení, které nejlépe vyhovuje jejich konkrétním předpokladům pro vývoj aplikací.

4.2.2 Srovnání podle funkcionality bezplatných verzí

Funkce	Mendix	Microsoft Power Apps	OutSystems
Webové aplikace	ano	ne	ano
Mobilní aplikace	ano	ano	ano
PWA	ano	ne	ano
Testovací aplikace	neomezeno	neomezeno	neomezeno
Sdílení aplikací	neomezeno	neomezeno	neomezeno
Publikace aplikací	neomezeno	ne	1
Velikost uložště	0,5 GB	3 GB	1 GB
Prostředí	Desktopové	Webové	Desktopové
Integrace	omezené	omezené	omezené
Podpora programovacích jazyků	HTML, CSS, Java, JavaScript, React a SQL	HTML, CSS, JavaScript, SQL a Power Fx	HTML, CSS, Java, OSL, JavaScript, SQL a C#
Propojení s vlastním API	ano	ano	ano
Šablony	ano	ano	ano
Responzivita	ano	ano	ano

Tabulka 4.4: Srovnání verzí zdarma low-code platforem podle funkcionality

Do srovnání podle funkcionality verzí zdarma nebyly zohledněny platformy Quickbase a ZohoCreator z důvodu, že ani jedna z platforem nemá svou verzi zdarma pouze několika denními trial, které by ve srovnání byly značně ve výhodě a do finálního výběru by trial nebyl vhodný pro podrobnější analýzu.

Mezi třemi posuzovanými platformami pro vývoj low-code aplikací Mendix, Microsoft Power Apps a OutSystems existují klíčové rozdíly ve funkčnosti. Zatímco Mendix a OutSystems podporují vytváření webových, mobilních aplikací a PWA [6, 8], Microsoft Power Apps se zaměřuje výhradně na mobilní aplikace [7]. V oblasti datázové kapacity je Mendix omezen na 0,5 GB, zatímco Power Apps a OutSystems nabízejí větší kapacitu s 3 GB a 1 GB.

Když se podíváme na programovací jazyky a možnosti integrace tak všechny platformy poskytují široké možnosti integrace, i když pro bezplatnou verzi omezené, a podporují programovací jazyky jako HTML, CSS, JavaScript a SQL, což dává vývojářům větší volnost při návrhu aplikací. Microsoft Power Apps rozšiřuje svůj seznam podporovaných jazyků o jazyk Power Fx, Mendix poskytuje navíc jazyk React a Java a OutSystems rozvíjí možnosti svých uživatelů využitím jazyků OSL, Java a C#. Všechny platformy umožňují propojení s vlastními API a poskytují šablony pro rychlý vývoj, dostupnost všech komponent a responzivitu aplikací.

Srovnání podle podpory bezplatných verzí

Platformy Mendix, Microsoft Power Apps a OutSystems poskytují podporu ve svých bezplatných verzích prostřednictvím aktivních komunit, které sdílejí řešení a tipy, a obsáhlým technickým dokumentacím. Tyto dokumentace jsou detailní a obsahují obrázky a videa, která lépe ilustrují, jak efektivně pracovat s danou platformou. Mendix navíc nabízí všem uživatelům přístup do své online akademie, kde si mohou rozšířit své dovednosti prostřednictvím vzdělávacích programů a získat certifikáty, nebo využít několik průvodců. OutSystems také poskytuje svou akademii, kde jsou k nalezení výukové lekce, certifikáty a průvodci. Uživatelé Microsoft Power Apps mohou nalézt kurzy pro prohloubení svých znalostí na Microsoft Learn.

4.3 Výběr tři nástrojů pro podrobnější analýzu

Pro podrobnější analýzu byly vybrány no-code nástroje Adalo, Glide a Softr, především kvůli jejich jednoduchosti a přístupnosti pro běžné uživatele, což je zásadní rozdíl oproti low-code platformám. No-code platformy minimalizují potřebu technických znalostí, což umožňuje uživatelům bez programátorského zázemí efektivně a rychle vytvářet aplikace. To je v příkrém kontrastu s low-code platformami, které ačkoli zjednodušují vývojový proces, stále vyžadují určité technické dovednosti a porozumění základním programovacím konceptům.

Glide vyniká především svou integrací s Google Sheets, která umožňuje běžným uživatelům transformovat tabulky na funkční aplikace s minimálním technickým zásahem. Adalo nabízí rozsáhlé možnosti pro design interakcí a vizuální přizpůsobení, které umožňují uživatelům tvořit esteticky přitažlivé aplikace bez nutnosti psaní kódu. Softr, se svými pokročilými šablonami a bloky navrženými pro integraci s Airtable, poskytuje rychlou cestu k vytváření robustních webových aplikací bez potřeby jakéhokoli programování.

Tyto platformy poskytují nástroje, které jsou intuitivně srozumitelné i pro nevyvojáře, což umožňuje širší skupině lidí pracovat na vývoji aplikací a digitálních

projektů. To přináší výhodu v podobě demokratizace vývoje softwaru, což je klíčovým faktorem ve stále více digitalizovaném světě, kde schopnost rychle reagovat na změny a implementovat nové nápady může přinést výraznou konkurenční výhodu. Tento přístup podporuje rozhodnutí zaměřit se na no-code platformy.

Ukázková aplikace pro porovnání nástrojů

5

Při výběru mezi rostoucím počtem no-code a low-code platformem je klíčové porozumět jejich schopnostem a omezením v praxi. Využití testovací aplikace jako nástroje pro analýzu a srovnání těchto platformem poskytuje cenné praktické informace, které přesahují teoretické specifikace a marketingové prezentace. Tento přístup umožňuje přímé porovnání funkčnosti, uživatelské přívětivosti a flexibility jednotlivých platformem, což zjednodušuje a zlepšuje proces srovnání a analýzy vybraných platformem. Tímto způsobem se testovací aplikace stává nezbytným krokem pro finální zhodnocení platformem.

Jako ukázková aplikace byla vybrána správa projektů pro porovnání nástrojů v rámci analýzy a hodnocení. Cílem je poskytnout ucelený pohled na to, jak různé platformy zvládají běžné vývojové úkoly a jaké mají silné a slabé stránky při budování aplikace pro správu projektů. Vybraná aplikace slouží k demonstraci klíčových funkcionalit, které tyto platformy nabízejí, a jejich schopnosti splnit potřeby uživatelů v oblasti efektivní správy projektů a úkolů. Cílovou skupinou jsou organizace a týmy, které potřebují jednoduché, ale flexibilní řešení pro organizaci svých projektů a úkolů.

5.1 Popis ukázkové aplikace

Správa projektů je moderní webová aplikace, která je navržena tak, aby uživatelům usnadnila organizaci a správu jejich pracovních projektů, čímž zvyšuje produktivitu. Aplikace by měla být kompatibilní s různými webovými prohlížeči a měla by se vyznačovat responzivním designem. Aplikace by měla umožňovat uživatelům vytvářet, spravovat a sledovat projekty, stanovovat jejich lhůty, vytvářet úkoly v daných projektech, což poskytuje komplexní přehled o všech aktivitách a povinnostech spojených s projektem. Hlavním cílem aplikace je zjednodušit proces plánování projektů a zvýšit přehled o nich.

Aplikace nabízí pokročilé možnosti pro víceuživatelské prostředí, kde každý uživatel může vytvářet projekty ve specifických kategoriích a vytvářet k nim úkoly. Tyto úkoly pak lze přiřazovat jednotlivým členům týmu. Každý uživatel má přístup k editaci pouze k projektům a úkolům, které vytvořil. Důležitým prvkem je, že uživatelé mohou úkoly nejen vytvářet a editovat, ale i označovat je jako splněné, což zlepšuje sledování pokroku práce.

Uživatelé mohou vytvářet a spravovat kategorie projektů a týmy, které mohou být dynamicky upravovány podle potřeb projektu. Přístup do aplikace je zabezpečený přihlašovacím systémem, který zajišťuje, že každý uživatel má přístup pouze k těm informacím a akcím, které jsou pro něj relevantní.

5.2 Databáze aplikace

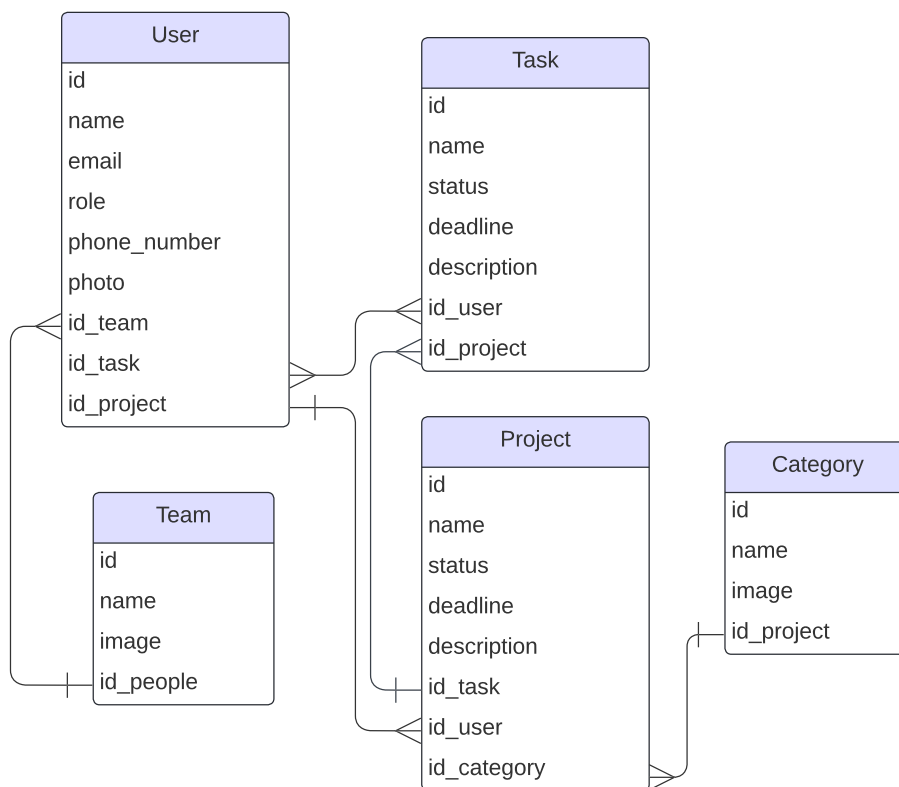
Navržená databáze pro správu projektů, která je vidět na obrázku 5.1, je základem pro aplikaci, která poskytuje strukturu pro správu a koordinaci projektů a úkolů v organizaci. Každý uživatel aplikace má svůj vlastní záznam v tabulce „User“, který obsahuje nejen základní osobní informace, ale i propojení s příslušnými týmy a projekty, ke kterým má přístup. Týmy jsou definovány v tabulce „Team“, což usnadňuje správu skupin lidí pracujících na různých projektech. Projekty a úkoly jsou pečlivě strukturovány v tabulkách „Project“ a „Task“, kde každý záznam obsahuje informace o stavu, deadline a dalších detailech, které jsou klíčové pro jejich sledování a úspěšné dokončení. Kategorie, definované v tabulce „Category“, poskytují další vrstvu organizace, která umožňuje třídit projekty do logických skupin a zjednodušuje navigaci a filtrování projektů podle oblasti. Tato databáze je navržena s důrazem na efektivní ukládání dat, zajištění integrity dat mezi tabulkami a poskytování rychlého přístupu k relevantním informacím pro uživatele aplikace.

5.2.1 Relace

Uživatel (User) a Tým (Team): Každý uživatel je součástí přesně jednoho týmu a jeden tým může zahrnovat mnoho uživatelů. To je relace 1:N mezi týmy a uživateli.

Projekt (Project) a Kategorie projektů (Category): Každý projekt patří pouze do jedné kategorie a jedna kategorie může obsahovat mnoho projektů. To je relace 1:N mezi kategoriemi projektů a projekty.

Uživatel (User) a Úkol (Task): Každý uživatel může mít přiřazeno mnoho úkolů a jeden úkol může mít přiřazeno mnoho uživatelů. To je relace M:N mezi úkoly a uživateli.



Obrázek 5.1: ER diagram pro Project management

Uživatel (User) a Projekt (Project): Každý projekt má jednoho uživatele jako svého vlastníka a uživatel může vytvořit mnoho projektů. To je relace 1:N mezi uživateli a projekty.

Projekt (Project) a Úkol (Task): Každý projekt může obsahovat mnoho úkolů a jeden úkol může být přiřazen pouze v jednom projektu. To je relace 1:N mezi projekty a úkoly.

Tyto relace umožňují aplikaci správy úkolů flexibilně spravovat úkoly, jejich přiřazení uživatelům a organizaci úkolů do kategorií. Role a oddělení umožňují další úroveň organizace a kontrolu přístupových práv uživatelů, zatímco struktura Správa Úkolů poskytuje detailní pohled na přiřazení, stav a plánování jednotlivých úkolů.

5.3 Use-case

Use-case, neboli případ užití je technika používaná v softwarovém inženýrství k popisu funkcionality systému z pohledu jeho koncových uživatelů. Pomáhá identifikovat, popsat a organizovat požadavky na systém. Případ užití vykresluje scénář, ve kterém uživatel (nebo jiný systém) interaguje se systémem k dosažení konkrétního cíle. Každý use-case obsahuje několik důležitých komponent, včetně názvu, popisu, aktérů (uživatelů nebo systémů, kteří s případem užití interagují), předpokladů, hlavního scénáře (kroků akce) a alternativních toků.

Use-case diagram, který je vidět na obrázku 5.2, představuje různé akce, které může uživatel provést v rámci aplikace pro správu projektů.

1. Registrace a Přihlášení

Registrace a Přihlášení: Uživatel se může zaregistrovat do systému nebo se přihlásit, pokud již má účet.

2. Správa Projektů:

Vytvořit/Upavit/Smazat Projekt: Uživatel může založit nový projekt, upravit existující nebo smazat projekt.

Vytvořit/Upavit/Smazat Úkol ve vašem Projektu: Uživatel může přidávat, měnit nebo odstraňovat úkoly uvnitř svých projektů.

3. Úkoly:

Označit Úkoly jako hotové: Uživatel má možnost označit úkoly za dokončené.

Připojit Další Uživatele k Úkolům: Uživatel může přiřadit úkoly ostatním uživatelům.

4. Uživatelský Profil:

Upravit/Smazat Profil: Uživatel může upravit informace ve svém profilu nebo svůj profil smazat.

5. Tým:

Přidat se k Týmu: Uživatel se může připojit k existujícímu týmu.

Vytvořit/Upavit/Smazat Tým: Uživatel může vytvořit nový tým, upravit nebo smazat tým.

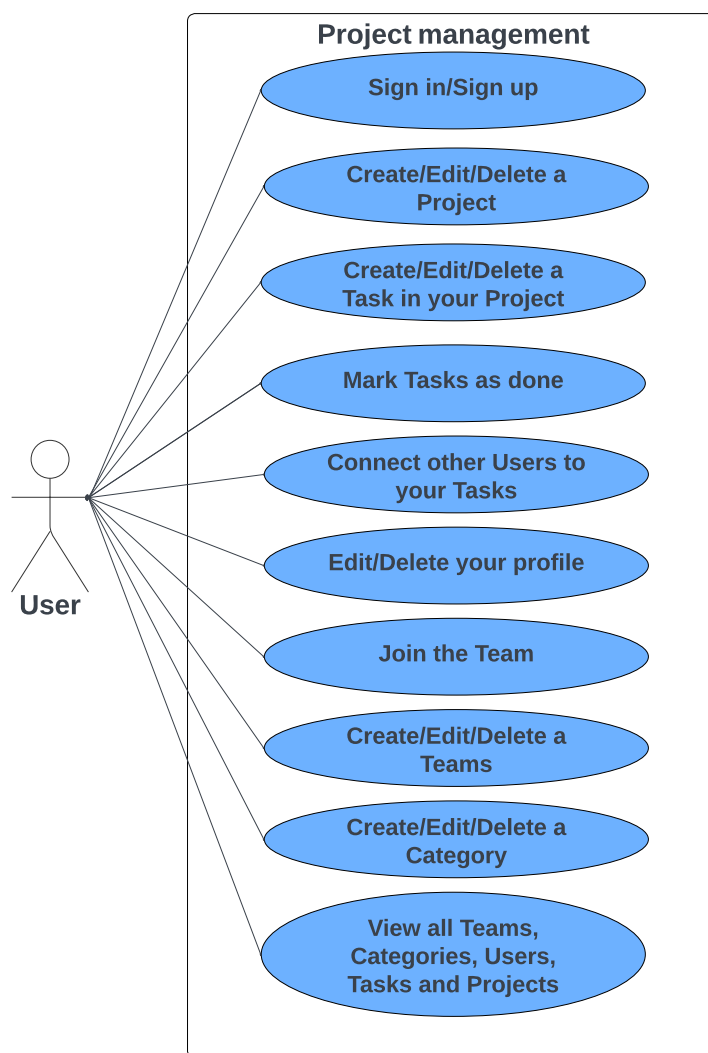
6. Kategorie:

Vytvořit/Upavit/Smazat Kategorii: Uživatel může přidat novou kategorii pro projekt, změnit nebo smazat stávající kategorii.

7. Přehled:

Zobrazit všechny Týmy, Kategorie, Uživatele, Úkoly a Projekty: Uživatel může získat přehled o všech aspektech aplikace, včetně týmů, kategorií, uživatelů, úkolů a projektů.

Tento diagram ukazuje rozsah interakcí, které může uživatel s aplikací provést, a poskytuje rámec pro pochopení, jak uživatelé aplikace komunikují s různými komponentami systému.



Obrázek 5.2: Use-case pro Project management

Porovnání použitelnosti a UX vybraných nástrojů

6

Tato kapitola se zaměřuje na hodnocení a srovnání použitelnosti a uživatelské zkušenosti tří vybraných nástrojů: Adalo, Glide a Softr. Cílem je identifikovat, které aspekty těchto platforem umožňují uživatelům nejen snadno a efektivně vytvářet aplikace, ale také které prvky mohou použitelnost omezovat či ztížit. Na základě ukázkového příkladu aplikace správa projektů, představené v předešlé kapitole, provedeme detailní analýzu klíčových faktorů, které ovlivňují celkovou uživatelskou spokojenost a efektivitu práce s těmito nástroji. Srovnání bude prováděno na základě použitelnosti nástrojů vytvářet aplikace a uživatelské zkušenosti, které jsou specifikovány v částech zaměřených na jejich hodnocení.

6.1 Popis vytváření ukázkové aplikace ve vybraných nástrojích

Pro porovnání no-code nástrojů bylo nezbytné detailně se zaměřit na proces vytváření ukázkové aplikace pro správu projektů v každém z vybraných nástrojů. Tento postup umožnil získat podrobný popis každé platformy, který sloužil jako základ pro následné hodnocení a porovnání. Každý z těchto nástrojů nabízí jedinečný soubor funkcí a přístupů k designu, které výrazně ovlivnily jak proces vývoje, tak konečný uživatelský zážitek. Postupně byly prozkoumány klíčové kroky vytváření aplikace, od návrhu databáze a uživatelského rozhraní přes implementaci specifických funkcionalit, jako jsou akční tlačítka a interaktivní prvky, a nakonec nasazení. Tato analýza nám umožnila nejen porozumět silným a slabým stránkám každého nástroje, ale také poskytla cenné vhledy do toho, jak no-code platformy mohou efektivně podporovat vývoj aplikací běžným uživatelem.

6.1.1 Vývoj databáze

V rámci vývoje databáze v no-code platformách Adalo a Glide probíhá tvorba databáze a aplikace společně v jednom integrovaném prostředí. Toto umožňuje bezproblémovou synchronizaci a interakci mezi databázemi a uživatelským rozhraním, což výrazně zjednodušuje celý vývojový proces. Naopak v Softr je situace odlišná, neboť databáze musí být vytvořena odděleně pomocí externího nástroje Airtable. Tento přístup vyžaduje, aby uživatelé spravovali data v Airtable a poté je integrovali do Softr, což přidává další krok do procesu vývoje aplikace, ale zároveň nabízí flexibilitu v pokročilejší manipulaci s daty.

Adalo

Vývoj databáze v Adalo byl proveden s cílem zajistit efektivní správu a strukturování dat pro aplikaci na správu projektů. Na počátku byly vytvořeny tabulky na základě návrhu ER diagramu pro správu projektů. Pro každou tabulku byly definovány specifické atributy s datovými typy, aby odpovídaly charakteristikám a požadavkům dat. Adalo nabízí datové typy jako text, číslo, logické hodnoty (true/false), datum a čas, samotné datum, obrázky, soubory a relace. Datový typ relace, fungující jako cizí klíč, byl zaveden pro propojení tabulek a umožnění vzájemné interakce dat. Typy relací, jako jsou 1:N, N:1 a M:N, byly zvoleny na základě struktury a potřeb aplikace, což umožnilo modelovat komplexní vztahy mezi daty. U vazby M:N není potřeba vytvářet propojovací tabulku, vazba je automaticky podporován přímo v rámci konfigurace relací mezi dvěma tabulkami. Při výběru relací v databázích jsou vždy doprovázeny textovým popisem jak je vidět na obrázku 6.1, který uživatelům vysvětluje, co každá relace znamená a jak funguje. Pro příklad relace mezi uživateli a projekty bylo vybrána relace 1:N. Toto pomáhá uživatelům lépe pochopit, jak jsou data propojena, a zajistit správné nastavení vztahů mezi tabulkami.

Aby uživatel mohl v Adalo úspěšně vytvořit a spravovat funkční databázi, je nezbytné, aby měl základní znalosti v oblasti tvorby a správy databáze, které zahrnují tvorbu tabulek, včetně jejich atributů, a relací mezi tabulkami. Dále je důležité rozumět různým datovým typům a jejich praktickému využití v rámci aplikace. Tyto datové typy jsou ale prezentovány jednoduchým a srozumitelným způsobem, což umožňuje i uživatelům bez velkých znalostí o datových typech snadno pochopit, jak je využít ve svých aplikacích. Pokud uživatel nemá dostatečné znalosti o tvorbě a správě databází, technická dokumentace Adalo ¹ nabízí jak stručný, tak podrobný popis procesu tvorby a správy databází. Kromě textových návodů je k dispozici také instruktážní video, které uživatelům poskytne vizuální představu o tom, jak správně databázi vytvořit a spravovat.

¹Technická dokumentace Adalo: <https://help.adalo.com/>

How are Users & Project related?



Obrázek 6.1: Popis relací v Adalo

Databáze byla nakonec naplněna daty prostřednictvím formulářů, které automaticky doplnily záznamy do tabulek, kde se pro každý záznam automaticky generuje ID. V attributech, které byly určeny jako relační (cizí klíče), bylo umožněno výběrem záznamů z příslušných tabulek. Obrázky byly nahrány do formulářů přímo z lokálního disku, což je jediná možnost, kterou Adalo nabízí.

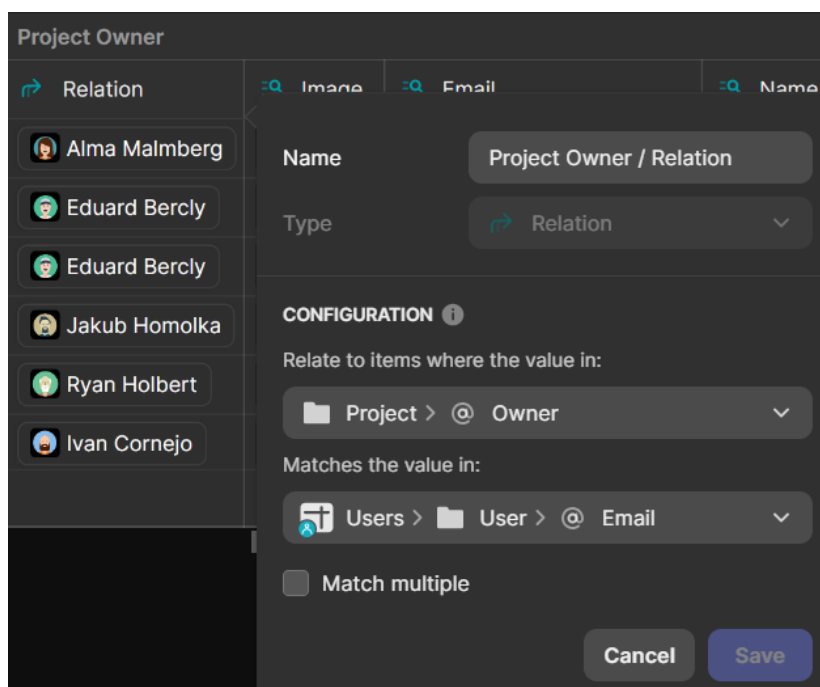
Vývoj databáze v Adalo byl velice snadný a rychlý, avšak vyžadoval od uživatele určité znalosti v oblasti databázového modelování. Pro ty, kteří s těmito dovednostmi nedisponují, je nezbytné se obrátit na technickou dokumentaci poskytovanou platformou. Tato dokumentace nabízí jak základní přehledy, tak podrobné instrukce, které uživatelům umožňují pochopit a efektivně využívat nástroje potřebné pro správnou tvorbu a správu databáze v Adalo.

Glide

Vývoj databáze pro aplikaci na správu projektů v nástroji Glide započal vytvořením tabulek v prostředí Glide Tables, které slouží jako základ pro ukládání a organizaci dat aplikace. Pro každou tabulku v Glide Tables byly vytvořeny sloupce, které představují atributy s příslušnými datovými typy. Glide nabízí širokou škálu datových typů. Na rozdíl od Adalo má Glide například datový typ e-mail, telefonní číslo, URL, emoji, tzv. počítané datové typy, kterých je mnoho, a datové typy získané různými integracemi. Specifické datové typy jako e-mail, telefonní číslo a URL blíže upřesňují, jak je uživatel může použít a tím mu usnadnit práci. Vzhledem k tomu,

že databázové prostředí Glide nepodporuje přímo vazby M:N, které jsou klíčové pro modelování vztahu mezi uživateli a úkoly, bylo nutné vytvořit speciální propojovací tabulku. Tato tabulka obsahuje cizí klíče, které odkazují jak na uživatele, tak na úkoly, a umožňuje tak efektivně sestavit požadovanou vazbu M:N. Tabulky byly poté ručně vyplněny daty, kde se také pro každý záznam vygenerovalo ID, pokud byl pro ně vytvořen sloupec, a obrázky byly nahrány z disku, ale lze je nahrát také z URL adresy.

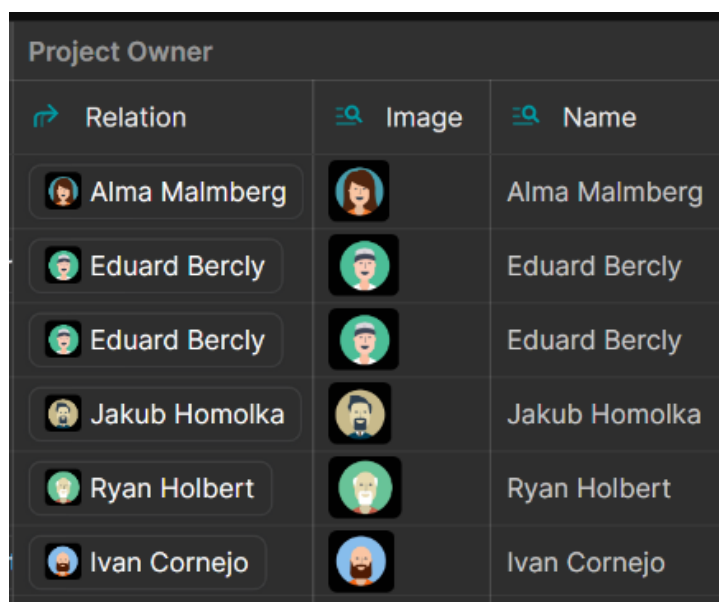
V prostředí Glide lze relace mezi tabulkami vytvořit sloupcem s funkcí „relation“, která je zařazena mezi počítané (computed) datové typy. Tyto sloupce pak reprezentují cizí klíče. Pro nastavení této relace je potřeba specifikovat sloupec v aktuální tabulce, který obsahuje identifikátor, a vybrat cílovou tabulku. V této tabulce je pak vybrán sloupec s odpovídajícími hodnotami, na základě kterých se vytvoří relace. Zde bylo třeba uvést zda se může připojit na jeden či více záznamů, podle toho jaká relace byla potřeba. V praxi je obvyklé používat pro propojení tabulek cizí klíč, který odkazuje na primární klíč v jiné tabulce. V Glide však při vytváření záznamů přes aplikaci uživatelé nemohou přímo vybírat ID, protože to je automaticky generováno a uživatelé obvykle nemají přehled o konkrétních ID jednotlivých entit a nástroj neumí na základě atributu vybrat ID příslušného záznamu. Kvůli této omezené dostupnosti ID bylo třeba využít jiného atributu entit jako identifikátory při vytváření relací.









Obrázek 6.2: Vytváření relace v Glide

Jako příklad lze uvést relaci v databázové tabulce projektů, kde je vytvořena relace mezi projekty a uživateli. Při vytváření relace byl jako identifikátor zvolen sloupec pojmenovaný „Owner“ v tabulce projektů, který obsahuje e-mail. Odpovídající hodnoty pro tuto relaci jsou v sloupci „Email“ v tabulce uživatelů, což umožňuje propojení projektů s jejich vlastníky, jak je vidět na obrázku 6.2. Nakonec nebyla zaškrtnuta možnost více záznamů, protože projekt má vždy pouze jednoho vlastníka. Tímto způsobem byla zajištěna funkčnost aplikace tak, že uživatelé mohou jednoduše vybírat a spojovat data na základě srozumitelných a známých názvů entit namísto neznámých ID čísel. Toto uspořádání pomohlo udržet aplikaci intuitivní a snadno použitelnou pro koncové uživatele.

Dále se v Glide relace mezi tabulkami netvoří přímo, jak je tomu například v Adalo, což omezuje možnosti přímého vyhledávání záznamů z jiných tabulek. Kvůli této limitaci bylo nutné využít funkci „lookup“, která umožňuje zobrazit konkrétní atributy z tabulek, s nimiž je vytvořena relace. Tato funkce zobrazuje pouze jeden atribut z každého záznamu, což znamená, že pro zobrazení více atributů z jedné spojené tabulky bylo třeba vytvořit více „lookup“ sloupců. Pro příklad jsou uvedeny „lookup“ sloupce z tabulky projektů, které zobrazují obrázek a jméno jednotlivých vlastníků projektů. Tyto sloupce jsou zobrazeny na obrázku 6.3. Tento proces je relativně časově náročný, jelikož každý z těchto sloupců musí být nastaven individuálně, aby odpovídal specifickým požadavkům aplikace.



Relation	Image	Name
Alma Malmberg		Alma Malmberg
Eduard Bercly		Eduard Bercly
Eduard Bercly		Eduard Bercly
Jakub Homolka		Jakub Homolka
Ryan Holbert		Ryan Holbert
Ivan Cornejo		Ivan Cornejo

Obrázek 6.3: Lookup sloupce pro vlastníka projektu v Glide

Vytváření databáze v Glide je obecně složitější než v Adalo, přičemž obě platformy vyžadují od uživatele základní znalosti databázového modelování. Klíčovým

rozdílem je, že v Glide musí uživatelé vytvářet propojovací tabulku pro realizaci vazby M:N, což může být pro některé uživatele náročné, zejména když nástroj ani technická dokumentace neposkytuje konkrétní pokyny k tomuto typu vazby. V dokumentaci jsou sice popsány jednodušší vazby jako 1:1, N:1 a 1:N, ale informace o M:N vazbách chybí.

Při vytváření relací mezi tabulkami Glide uživatele neupozorňuje na nutnost dalších kroků. Například, že je potřeba pro zobrazení dat z propojených tabulek použít funkci „lookup“. Uživatelé se musí dozvědět (buď z technické dokumentace, fóra, nebo výukových videí), že je nutné vytvořit speciální „lookup“ sloupce pro každý atribut, který chtějí z propojené tabulky zobrazit. Tento proces může být pro uživatele bez technického pozadí obtížný a časově náročný, jelikož vyžaduje více manuální konfigurace.

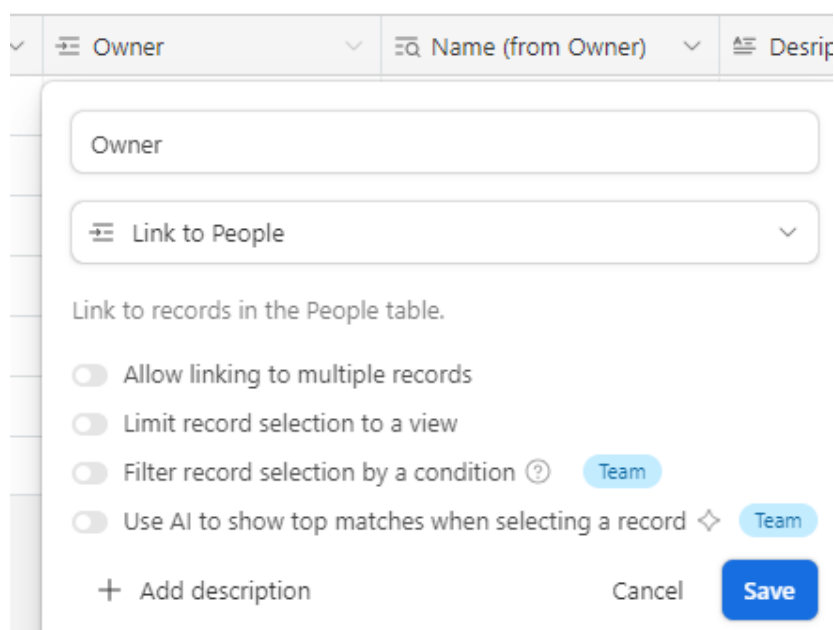
Celkově je zřejmé, že i když se Glide snaží proces tvorby databáze co nejvíce usnadnit, mnoho důležitých informací a návodů si musí uživatelé najít sami, což může vést k frustraci a potenciálně omezenému využití platformy pro složitější databázové operace.

Softr

Vývoj databáze v Softr musel být realizován pomocí připojení tabulek z externí databáze, konkrétně z Airtable, protože Softr neposkytuje své prostředí pro správu dat. V nástroji Airtable byly nejprve vytvořeny tabulky reprezentující entity aplikace a v nich sloupce reprezentující atributy se správnými datovými typy. Stejně jako Glide, tak i Softr nabízí specifické datové typy, jako jsou e-mail, URL a telefonní číslo. Tyto přesně definované datové typy usnadňují uživatelům práci, neboť přesně specifikují očekávaný formát dat a zvyšují tak efektivitu a přesnost při manipulaci s daty. Poté byly ručně do tabulek vyplněny data a obrázky, které lze nahrát z disku, onedrive, z URL adresy, webkamery, google disku a dropboxu.

Relace v Airtable byly vytvořeny funkcí „link to the another records“ ve sloupci, které představují cizí klíče, tedy odkaz na jinou tabulku. U každé tak vytvořené relace se nastavovalo, ke které tabulce se má připojit, a zaškrtovalo se, zda se jedná o odkaz na jeden nebo více záznamů. Pro příklad v tabulce projektů byl vytvořen sloupec s názvem „Owner“, označující vlastníka projektu. U tohoto sloupce byla nastavena funkce „link to the another records“ a byla vybrána tabulka uživatelů nazvaná „People“, jak lze vidět na obrázku 6.4. U této relace nebyla zaškrtnuta možnost více záznamů, protože projekt může mít pouze jednoho vlastníka. V tabulce uživatelů je pak automaticky vytvořen sloupec se stejnými vlastnostmi, ale připojení na tabulku projektů. U tohoto sloupce bylo nutné zaškrtnout možnost více záznamů, aby uživatel mohl vlastnit více projektů. Pro správné fungování aplikace bylo nezbytné upravit primární klíče (ID), které jsou nyní formátovány tak, aby obsahovaly

název entity a byly jedinečné díky číselné sekvenci, která se s každým novým záznamem zvyšuje. Tato změna byla provedena, aby se zajistila funkčnost cizích klíčů, které v rámci relací odkazují na tato ID v jiných tabulkách. Tento způsob propojení tabulek pomocí cizích klíčů je standardní a nedá se změnit. V nástroji Softr například pro komponenty výběru nebo filtrace entit lze použít pouze primární klíče, proto musela být provedena tato úprava. Stejně jako u Glide tak i v Airtable nejsou relace přímé a i zde byly vytvářeny sloupce s funkcí „lookup“ pro zobrazení záznamů z jiné tabulky a ani zde není na tuto nutnost uživatel nijak upozorněn a musí na to přijít sám.



Obrázek 6.4: Vytváření relací v Airtable

Po dokončení databáze v Airtable následovalo připojení této databáze k Softr. Softr umožňuje snadné připojení k Airtable prostřednictvím API nebo přímé integrace, což zajišťuje, že všechna data v Airtable jsou dostupná a aktualizovaná v reálném čase v aplikaci vytvořené v Softr.

Vývoj databáze v Airtable je oproti Glide jednodušší, hlavně díky tomu, že pro vazbu M:N není nutné vytvářet propojovací tabulku. Na druhou stranu, repetitivní proces vytváření „lookup“ sloupců je zdoluhavější než práce v Adalo, i když se „lookup“ sloupce vytvářejí jednoduše kliknutím na sloupec, který obsahuje relaci, a následným výběrem požadovaného atributu, který si přejeme zobrazit. Na rozdíl od Adalo, Airtable neumožňuje tak pohodlnou práci s relacemi ve smyslu jejich přímého použití v aplikacích. Pro zajištění funkčnosti aplikace, kde je například uživatel přiřazován k úkolu, není v Airtable možné jednoduše vybírat pouze z ID

uživatelů, které je v relaci použito bez možnosti úpravy. Toto ID, které se v Airtable automaticky zvyšuje s každým záznamem, by samo o sobě nebylo dostatečně informativní. Proto je obvykle nutné k tomuto ID přidat i jméno daného uživatele, aby bylo zřejmé, koho ID reprezentuje, jak je znázorněno na obrázku 6.5. Tímto způsobem je v Airtable modifikováno ID, aby neslo další informace, což usnadňuje uživatelům správnou orientaci a výběr v kontextu aplikace.



<input type="checkbox"/>	f_x ID	
<input type="checkbox"/>	2	Eduard Berclý
2	3	Ivan Cornejo
3	4	Fatima Delgadillo
4	5	Ryan Holbert
5	6	Jakub Homolka
6	7	Karel Homolka
7	8	Sophia Rodriguez
8	9	Alma Malmberg

Obrázek 6.5: Sloupec ID v tabulce uživatelů v Airtable

Stejně jako v Adalo a Glide, i pro práci s Airtable je nezbytné mít základní znalosti o modelování dat. V případě, že si uživatel není jistý, jak postupovat, může se obrátit na technickou dokumentaci Airtable, která poskytuje cenné informace o tom, jak správně vytvářet relace mezi tabulkami nebo vybírat vhodné datové typy pro specifické atributy. Celkově je Airtable snadný na použití a uživatel se s ním lehce naučí pracovat.

6.1.2 Vývoj uživatelského rozhraní (UI) a designu

Adalo

Vývoj uživatelského rozhraní a designu pro aplikaci na správu projektů v Adalo začal od základních stránek, které systém automaticky generuje, včetně úvodní stránky s přivítáním uživatele a stránek pro přihlášení a registraci. Všechny tyto stránky se pak dají upravit podle potřeb uživatele. Prvním krokem bylo vytvoření domovské stránky, která zahrnovala horní navigační panel. Tento panel měl reprezentovat

všechny hlavní entity z databáze, ale kvůli omezení Adalo, které umožňuje maximálně pět prvků v navigačním panelu, se kategorie vešly jen částečně a některé z nich musely být vynechány.

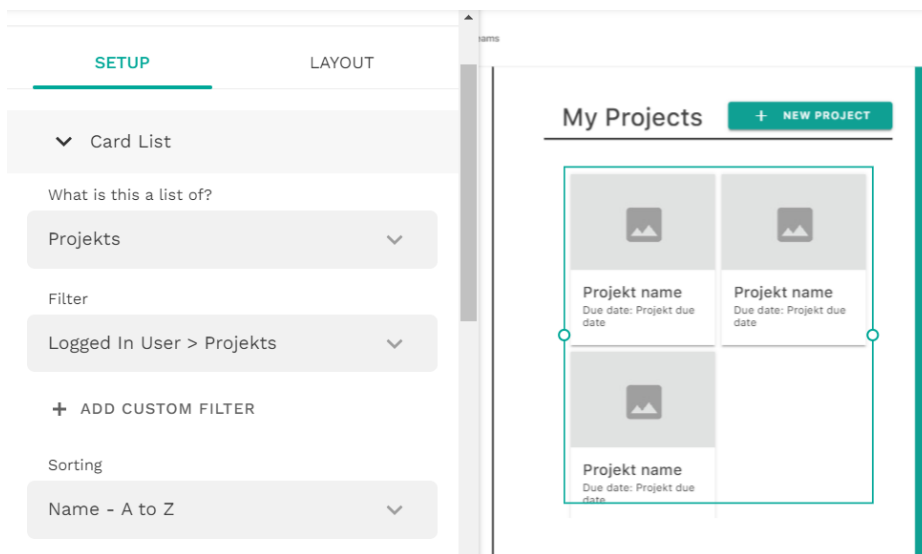
Horní panel na domovské stránce také obsahoval odkaz na uživatelský profil na pravé straně a na levé straně logo nebo obrázek sloužící jako titulek. Tělo stránky bylo postupně vyvíjeno přidáváním různých komponent metodou „drag and drop“, což umožňovalo flexibilní umístění prvků podle potřeb designu. Každá komponenta nabízela široké možnosti pro individuální nastavení designu, včetně barev, písma, rozměrů a dalších atributů. Toto individuální nastavení bylo provedeno formou „vyber a klikni“, kde jednotlivé parametry si uživatel volí z předem určených možností.

Pro každou entitu, s výjimkou kategorií, byla mnou vytvořena samostatná stránka, která kopírovala design horního panelu z domovské stránky. Tyto stránky byly vybaveny komponentami pro zobrazení seznamů, které prezentovaly záznamy z databáze. Součástí byl také často vyhledávací panel, filtrace dat podle specifických kategorií, a tlačítko pro otevírání oken s formulářem pro přidávání nových záznamů. U všech komponent využívajících data bylo nezbytné specifikovat, z které tabulky mají data čerpat. Dále bylo nastaveno, že po kliknutí na prvek v seznamu se uživatel dostane na detailní stránku, která nabízí kompletní informace o vybraném prvku, včetně možností pro jeho editaci a smazání.

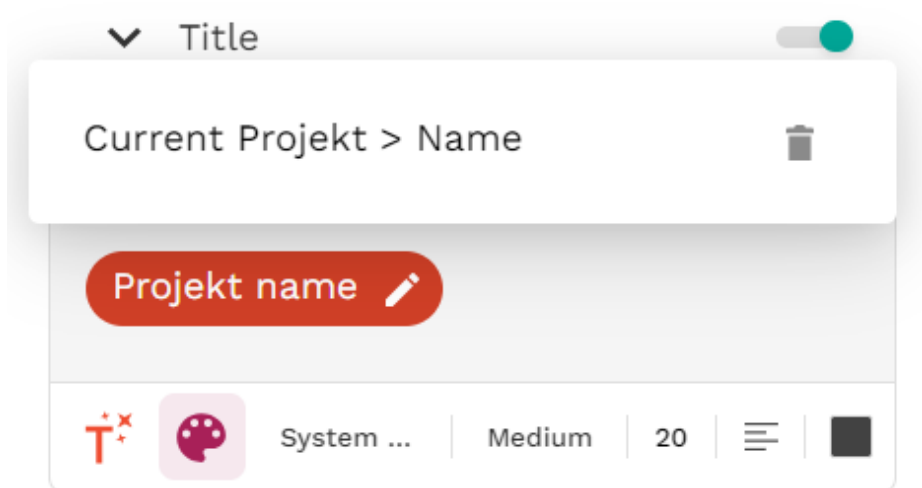
Pro příklad na domovské stránce byly vytvořeny dva seznamy, umístěné každý na opačné straně stránky, přičemž každý z nich sloužil odlišnému účelu. První seznam obsahuje projekty přihlášeného uživatele, zatímco druhý zobrazuje jeho úkoly. Pro oba seznamy byl vytvořen nadpis a pro seznam projektů bylo navíc přidáno vytvářecí tlačítko. Vytvářecí tlačítko umožňuje otevřít okno s formulářem pro vytvoření nového projektu, které bylo v aplikaci vytvořeno až později. Data pro oba seznamy jsou čerpána z příslušných tabulek, které byly nastaveny společně s filtry tak, aby zobrazovaly pouze položky patřící přihlášenému uživateli. Tímto způsobem je zajištěno, že uživatel vidí pouze položky, které jsou pro něj relevantní. Detailní nastavení seznamu projektů je ilustrováno na obrázku 6.6.

Dále bylo u obou seznamů nastaveno, aby se položky zobrazovaly v abecedním pořadí podle jejich názvů a aby se při kliknutí na jakýkoliv prvek v seznamu uživatel dostal na detailní okno, které poskytuje více informací o daném prvku. Tato okna s detaily byla vytvořena později během vývoje aplikace. Kromě toho bylo pro oba seznamy specifikováno, které atributy entit se mají zobrazovat, například jména projektů nebo úkolů. Konkrétní nastavení titulku pro seznam projektů je vizuálně zobrazeno na obrázku 6.7.

Stránky s detaily jednotlivých prvků byly vytvořeny s jednoduchým horním panelem obsahujícím pouze tlačítko pro skok na domovskou stránku a tlačítko zpět. Obsah stránky byl vyplněn všemi informacemi daného prvku a prvky s ním spojené, jako například v detailu projektu byl vložen název, deadline, vlastník, kategorie



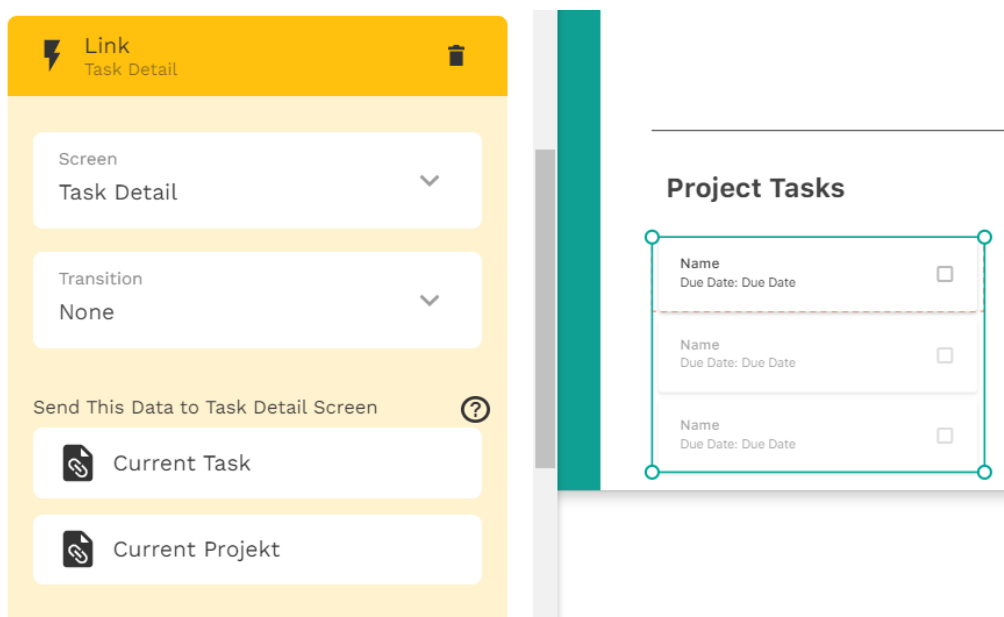
Obrázek 6.6: Nastavení seznamu projektů na domovské stránce v Adalo



Obrázek 6.7: Nastavení titulku prvku v seznamu

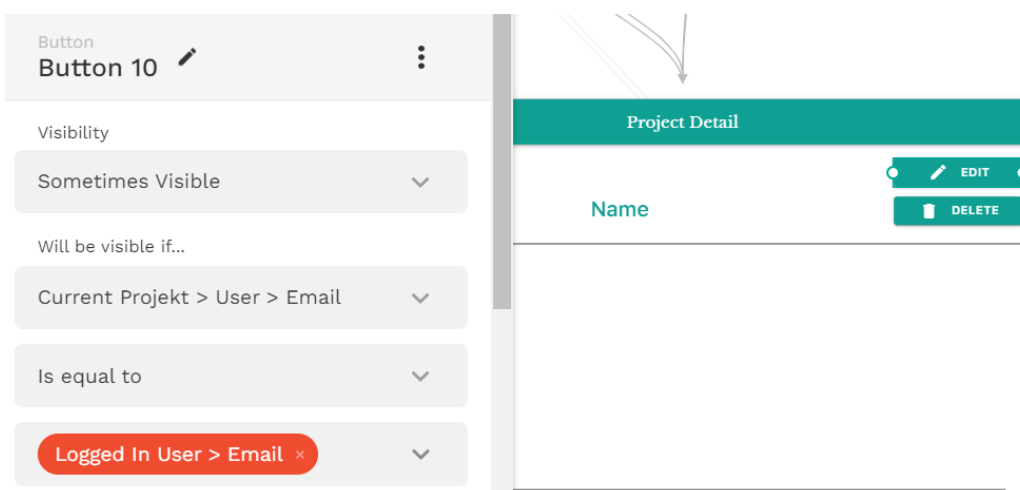
a seznam úkolů patřící do projektu. Na prvky vlastník, kategorie a úkoly bylo také nastaveno prokliknutí na detail daného prvku viz obrázek 6.8. Stránky pro přidání nového prvku a editaci stávajícího prvku sdílely design se stránkami s detaily prvku a obsahovaly formulář, který obsahoval atributy daného prvku, kde bylo možné vyplnit nebo upravit informace o prvku.

Tlačítka jsou uživatelem konfigurována tak, aby prováděla specifické akce, jako je přesměrování na jinou stránku nebo mazání záznamů z tabulek, a umožňovala



Obrázek 6.8: Nastavení linku pro seznam úkolů

nastavení vzhledu a viditelnosti, která podporuje nastavení přístupových práv. Viditelnost komponent je klíčová pro správné fungování aplikace, neboť zajišťují, že každý uživatel může mazat, upravovat a přidávat prvky, k nimž má oprávnění. Pro lepší představu nastavování viditelnosti komponent viz obrázek 6.9, který zobrazuje nastavení přístupových práv u editačního tlačítka na stránce detailu projektu. Aby bylo zajištěno, že editovat projekt může pouze vlastník daného projektu.



Obrázek 6.9: Nastavení přístupových práv u editačního tlačítka

Jako poslední byla vytvořena stránka uživatelského profilu, která poskytuje informace o uživateli a obsahuje formulář pro úpravu těchto informací a změnu hesla.

Vytváření uživatelského rozhraní v Adalo je přístupné díky „drag and drop“ editoru, což uživatelům umožňuje snadno prototypovat a upravovat aplikace bez nutnosti psát kód. Základní znalosti designu uživatelského rozhraní jsou však důležité, neboť Adalo nabízí velkou míru designové flexibility. To může být výhodou pro zkušené designéry, ale pro ty bez designových zkušeností může vést k vytvoření nepřehledných a nepřívětivých aplikací pro koncové uživatele. Nastavení různých vlastností a funkcionalit komponent v Adalo je značně zjednodušeno díky intuitivnímu systému „vyber a klikni“. Uživatelé jednoduše vybírají požadovanou funkcionalitu z výběrového seznamu a kliknutím nastavují specifické parametry. Tento přístup umožňuje rychlou a přehlednou konfiguraci komponent, aniž by bylo od uživatele vyžadováno rozsáhlé technické znalosti.

Glide

Vývoj uživatelského rozhraní a designu v nástroji Glide probíhá odlišně od Adalo, jelikož Glide používá funkci „drag and drop“ trochu jinak. Pracuje s výběrem komponent z předdefinované nabídky a jejich uspořádáním v rámci stránky v tzv. kontejnerech. Tento přístup omezuje flexibilitu v návrhu, protože uživatelé mohou pouze vybírat z dostupných komponent a definovat jejich pořadí. U komponent uživatelé nemají možnost přizpůsobení velikosti a umístění jednotlivých prvků tak volně jako v Adalo.

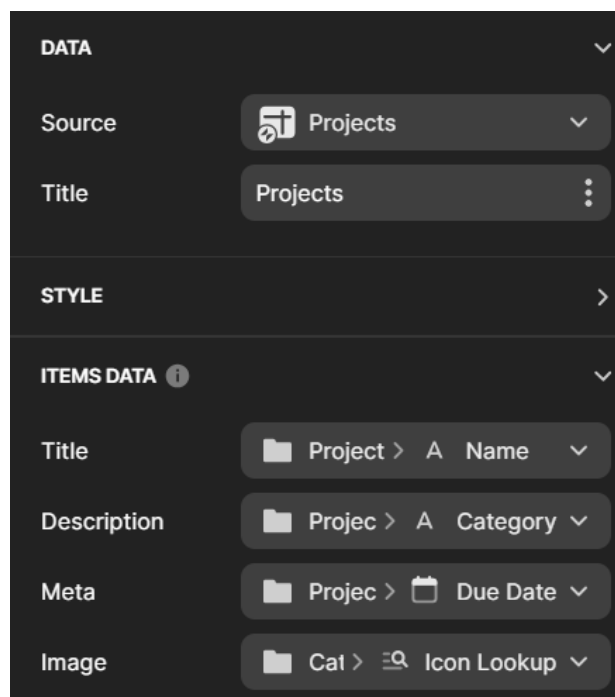
V nástroji Glide bylo nejprve nástrojem vygenerováno přihlašovací okno a uživatelé tak nemají možnost provádět jeho úpravy, což omezuje možnosti přizpůsobení designu. To samé se dá říct o bočním navigačním panelu, který sloužil jako hlavní navigační prvek. U navigačního panelu se dá nastavit pouze zda se bude jednat o boční či horní panel. Do navigačního panelu byly poté automaticky nástrojem přidávány odkazy na hlavní stránky, které byly vytvářeny až později. Pro celou aplikaci byla také stanovena barevná paleta, která zajistila vizuální konzistenci. Dále byly přidány titulek a obrázek, které definovaly vizuální identitu aplikace. Následně byly mnou vytvořeny jednotlivé hlavní stránky, které reprezentovaly všechny entity z databáze, včetně domovské stránky a stránky profilu uživatele. Všechny tyto stránky byly následně automaticky integrovány do bočního navigačního panelu, čímž byl uživatelům poskytnut snadný přístup k jednotlivým sekcím aplikace.

Obsah každé stránky byl vytvořen tak, aby odpovídal specifickým potřebám aplikace. Komponenty byly vybírány a řazeny do kontejnerů podle potřeby, což umožnilo efektivní organizaci a zobrazení informací v aplikaci. Každá komponenta měla možnost individuálního nastavení vzhledu, funkcí, akčních tlačítek, zobrazení a zdroje dat, pokud komponenta pracovala s daty z databáze. Pro komponenty zob-

razující seznamy prvků z tabulky byly nastaveny akce jako vytváření nového prvku, mazání a editace. Tyto akce byly automaticky integrovány do uživatelského rozhraní aplikace u komponenty, ve které byly vytvořeny.

Kromě toho byla nastavena viditelnost pro různé uživatele, aby bylo zajištěno, že uživatelé mohou interagovat pouze s prvky, ke kterým mají oprávnění. Podokna pro editaci a vytváření prvků byla navržena jako vyskakovací okna, přičemž editační okna obsahovala předvyplněná data existujícího prvku. Detaily jednotlivých prvků v seznamu byly prezentovány jako klasické stránky, které poskytovaly všechny informace o daném prvku a umožňovaly navigaci k dalším souvisejícím detailům. U editovacích a mazacích tlačítek v těchto detailech byla také nastavena viditelnost.

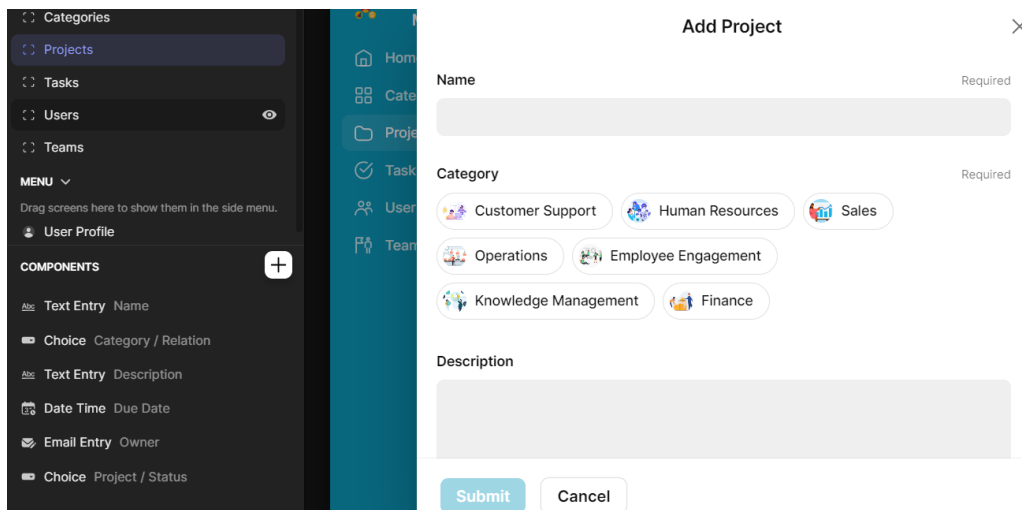
Pro příklad na stránce projektů byly vytvořeny dva kontejnery. Uvnitř prvního byl vytvořen kartový seznam všech projektů a ve druhém kanbanový seznam všech projektů. U kartového seznamu byl nastaven titulek seznamu a z jaké tabulky má brát data. Nakonec byl nastaven titulek, popis, metadata a obrázek u jednotlivých karet. U kterých bylo nastaveno z jakého sloupce v tabulce mají brát data viz obrázek 6.10.



Obrázek 6.10: Nastavení kartového seznamu projektů v Glide

Dále bylo možné u seznamu nastavit různé designové prvky pro karty v seznamu a aktivovat prvek vyhledávací lišty, která byla zaškrtnuta. Pro každý seznam byly dále definovány akce, přičemž první z nich umožňovala přidávání nových projektů skrze vyskakovací okno s formulářem. V tomto okně byly přidány komponenty,

kteří společně tvořily formulář pro vytvoření nového prvku, jak je ilustrováno na obrázku 6.11. U každé komponenty formuláře bylo specifikováno, do kterého sloupce v tabulce se mají data zapisovat, a bylo nastaveno, zda je vyplnění komponenty povinné. Pokud nebyla povinná komponenta vyplněna, dokončení akce nebylo možné.

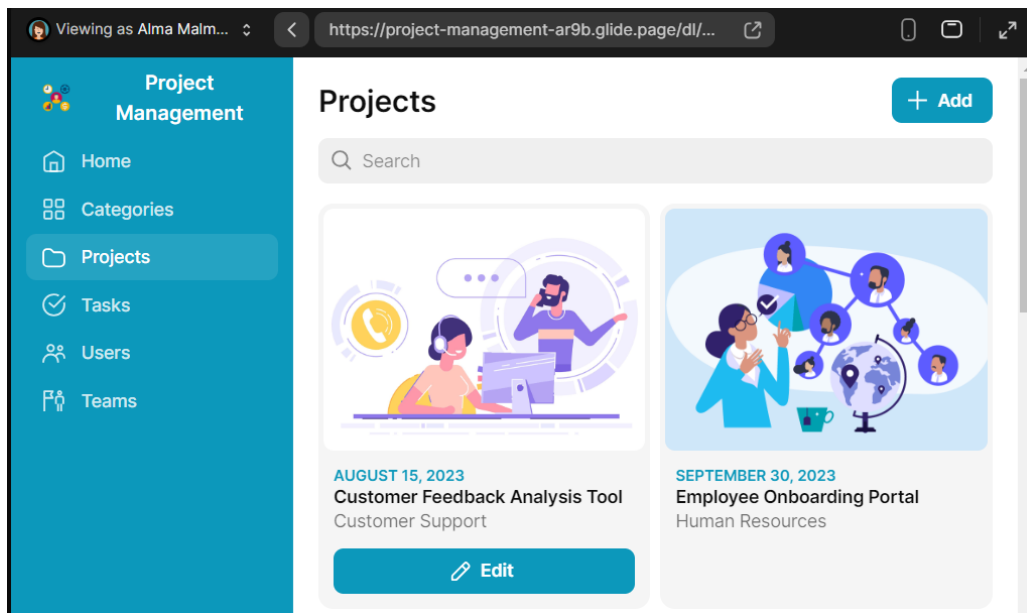


Obrázek 6.11: Vytváření formuláře pro přidání projektu v Glide

Když se vrátíme zpátky k nastavování akcí u kartového seznamu, byla dále nastavena akce pro kliknutí na jednotlivé karty, která otevírala detail daného projektu. Tento způsob linkování stránek je velmi podobný jako u Adalo. Nakonec bylo vytvořeno editační tlačítko v každé kartě seznamu, kde byla nastavena viditelnost pouze pro majitele projektu. Viditelnost komponent se nastavují úplně stejným způsobem jako v Adalo. Všechny akce reprezentované tlačítky se automaticky nástrojem vygenerovaly do výsledné stránky, která je vidět na obrázku 6.12.

Navzdory tomu, že interaktivní prvky komponent v Glide, jako jsou tlačítka, mají přednastavené styly a velikosti, které nelze modifikovat, uživatelé mohou upravovat základní atributy, jako jsou barvy, texty a ikony. Toto omezení sice zajišťuje konzistentní vzhled napříč celou aplikací, ale současně omezuje možnosti pro detailnější personalizaci a přizpůsobení designu.

Glide se zaměřuje na rychlost a efektivitu v procesu tvorby uživatelského rozhraní, čímž nabízí ideální řešení pro projekty, které vyžadují rychlé prototypování a nasazení bez hlubokých technických znalostí v oblasti designu narozdíl od Adalo. Platforma poskytuje vedení s určitými omezeními v designu, což uživatelům usnadňuje vytvoření funkčního a esteticky přijatelného designu. Tento přístup je zvláště vhodný pro týmy a jednotlivce, kteří chtějí rychle spustit aplikaci s omezenými zdroji pro customizaci a omezenými designovými schopnostmi.



Obrázek 6.12: Výsledná stránka projektů v Glide

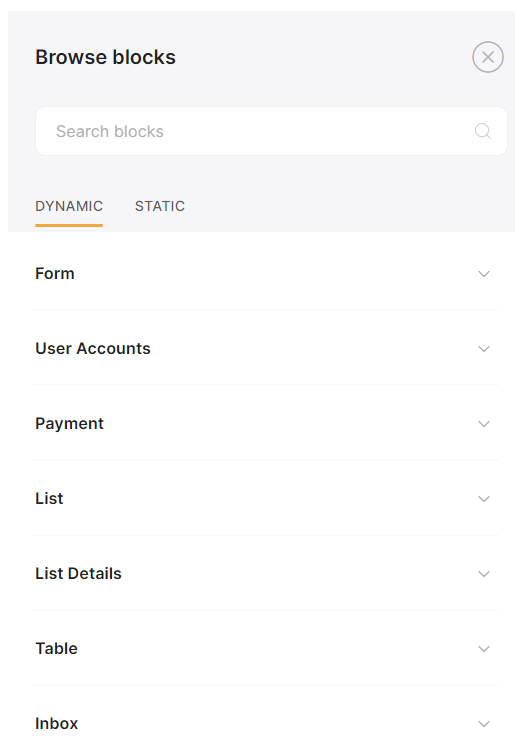
Porovnáním s Adalo, kde je možné vidět všechna okna najednou a uživatelé mohou snadno přecházet mezi nimi, Glide vyžaduje proklikávání mezi jednotlivými okny prostřednictvím vizualizace aplikace uprostřed obrazovky. Toto může poněkud komplikovat udržení celkového přehledu o aplikaci. Avšak, Glide nabízí výhodu v tom, že umožňuje uživatelům testovat funkcionality přímo v prostředí nástroje bez nutnosti přecházet do samostatného náhledu. To znamená, že jakmile je nějaká funkcionality dokončena, může být okamžitě vyzkoušena a validována, což urychluje proces vývoje.

Softtr

Vývojové prostředí Softtr nabízí omezenější možnosti nastavení designu jednotlivých prvků uživatelského rozhraní jako Glide. Proces vývoje začal vytvořením úvodní stránky, která zahrnovala horní navigační panel přítomný na všech ostatních stránkách aplikace. Boční panel, obsahující logo aplikace, poskytoval navigaci na hlavní stránky, které reprezentují entity v databázi, domovskou stránku a profil uživatele.

Jednotlivé stránky se vytvářejí v pravém horním rohu, kde se jen klikne na tlačítko nová stránka a zadá se název. Navigační panel je nástrojem automaticky vložen do všech stránek. Obsah stránek v Softtr je organizován do bloků, které jsou řazeny za sebou. Každý blok může obsahovat různý obsah, jako jsou seznamy, tabulky, formuláře nebo detaily prvků. Bloky jsou kategorizovány jako dynamické nebo statické,

v závislosti na tom, zda využívají data z databáze (viz obrázek 6.13). Při vytváření aplikace byly využívány především dynamické bloky.

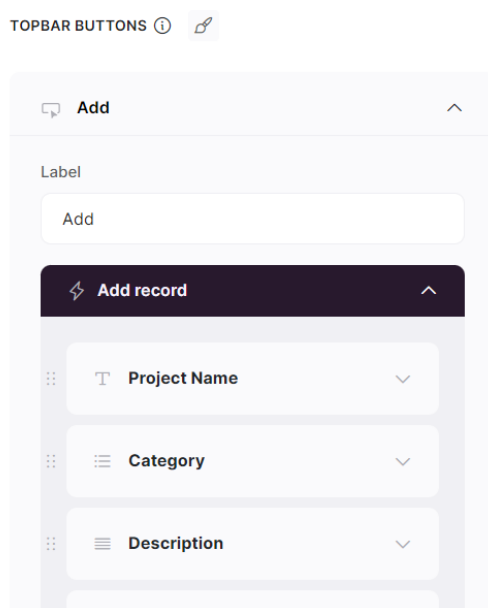


Obrázek 6.13: Nabídka bloků v Softr

Pro každý blok, který využívá data z databáze, bylo nezbytné nastavit zdroj dat, včetně specifikace, ze kterého nástroje, databáze a tabulky se mají čerpat data. Navíc bylo možné u každého bloku upravit design, nastavit filtry, specifikovat zobrazované informace, přidat akční prvky a definovat viditelnost, která jsou k dispozici pouze v placené verzi aplikace. Toto je limitující faktor ve srovnání s nástroji jako Adalo a Glide. Všechna tato nastavení jsou si velmi podobná s nástrojem Glide, ale Softr poskytuje trochu větší flexibilitu designu. Například s různými tlačítky, seznamy, či nadpisy se dají odsazovat, nastavovat velikost písma atd., což v Glide nelze.

V bloku reprezentujícím seznam prvků z databáze byla přidána tlačítka pro vytváření nových prvků a odkazy na stránky s detaily těchto prvků. Tyto tlačítka se také nastavují velmi podobně jako u nástroje Glide. Když se stiskne tlačítko pro vytváření, zobrazí se vyskakovací okénko s formulářem. Tento formulář se nastavuje v nastavení tlačítka, viz obrázek 6.14, což je rozdíl oproti Glide, kde se s vyskakovacím okénkem pracovalo jako se samostatnou stránkou. Tlačítka s odkazy na jiné stránky se tvoří úplně stejně jako v Adalo či Glide.

Po vytvoření stránek s detaily jednotlivých prvků byl do nich nejprve umístěn blok s detailem prvku, kde byly zobrazeny všechny důležité informace a akční tla-

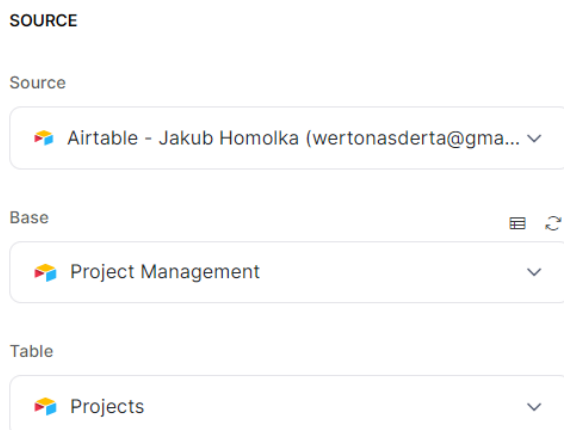


Obrázek 6.14: Vytváření formuláře pro přidání projektu v Softr

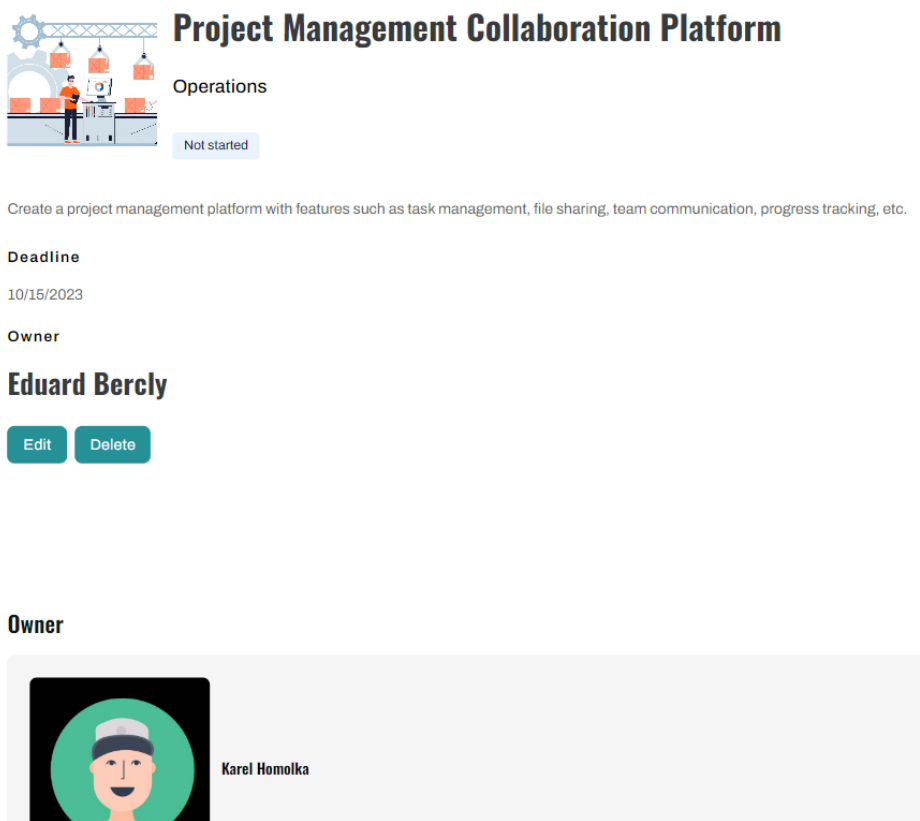
čítka pro editaci a mazání prvku. Editační formuláře byly vytvořeny a nastaveny stejně jako ty vytvářecí, což umožňuje efektivní interakci bez nutnosti opustit aktuální kontext uživatele.

Pro příklad u detailu projektu byly vytvořeny tři bloky: jeden s detailem na projekt, druhý seznamem uživatelů, kteří jsou majiteli projektu, a třetí se seznamem úkolů patřících do projektu. V prvním bloku s detailem se nejprve doplnily údaje o tom, z jaké tabulky, v jaké databázi a v jakém nástroji má brát data (viz obrázek 6.15). Poté byly připojeny jednotlivé sloupce k prvkům zobrazeným v detailu. Některé prvky jsou v bloku již připraveny, ale mohou být odstraněny nebo přidány jiné. Nakonec byla nastavena editační a mazací tlačítka. U mazacího tlačítka bylo ještě potřeba nastavit, kam se po smazání prvku přesunout, protože když se smaže daný prvek, jeho detail již nebude dostupný, a proto je potřeba přeskočit na jinou stránku. Ve výchozím nastavení je domovská stránka. V druhém bloku bylo nastaveno, že bude brát data z tabulky uživatelů a byl nastaven filtr, aby ukazoval pouze uživatele, kteří jsou majiteli daného projektu. Nakonec bylo nastaveno, aby se v seznamu zobrazovala pouze fotka a jméno uživatele a aby při kliknutí na uživatele v seznamu se přešlo na stránku s jeho detailem. V třetím bloku se seznamem úkolů bylo také třeba nastavit, odkud má brát data. Poté bylo nastavení stejné jako u nástrojů Glide a Adalo, zde se pouze nastavila podmínka aby se zobrazovaly pouze úkoly z daného projektu. Výsledná stránka je k vidění na obrázku 6.16.

Nakonec byly vytvořeny stránky pro přihlášení, registraci, obnovení hesla a uživatelský profil. Tyto stránky byly vytvořeny velmi snadno, protože Softr poskytuje



Obrázek 6.15: Připojení dat k bloku v Softr



Obrázek 6.16: Stránka s detailem projektu v Softr

bloky, které vždy odpovídají funkcionalitě dané stránky. Uživatel si pak jenom může nastavit design jednotlivých bloků nebo daný blok trochu upravit.

Vývoj uživatelského rozhraní v nástroji Softr poskytuje efektivní platformu pro rychlé sestavování aplikací, i když neumožňuje tak rozsáhlé možnosti designu jako například Adalo. Přestože je flexibilita v přizpůsobení designu a interaktivních prvků omezenější, Softr kompenzuje tuto nevýhodu dobře organizovaným systémem bloků, který zjednodušuje proces tvorby a umožňuje rychlé nasazení aplikace. Tento přístup nevyžaduje od běžných uživatelů pokročilé znalosti v oblasti designu, protože výchozí nastavení bloků je intuitivní a usnadňuje efektivní vývoj uživatelského rozhraní.

Srovnatelně s nástrojem Glide, Softr nabízí podobné funkce s mírně větší flexibilitou v designu, což může být výhodou pro uživatele hledající určitou míru kreativní svobody, ale stále potřebující rychlé a jednoduché řešení pro vývoj aplikace z dat. Celkově je Softr vhodný pro ty, kteří chtějí rychle vytvořit funkční aplikaci s profesionálním vzhledem bez potřeby hlubokého ponoru do složitostí uživatelského designu.

6.1.3 Nasazení

Adalo

V bezplatné verzi Adalo nelze aplikace publikovat, pouze je možné je sdílet zasláním URL adresy, což umožňuje uživatelům přístup přímo ve webovém prohlížeči. Toto řešení je vhodné pro sdílení v testovací fázi nebo pro získání zpětné vazby od omezeného počtu uživatelů. Pro přístup do aplikace je vyžadováno přihlášení. Uživatelé, kteří ještě nemají založený účet, musí projít procesem registrace. Přestože sdílení přes URL je pohodlné, pro komerční nasazení a širší distribuci je lepší zvolit placenou verzi Adalo, která nabízí rozšířené možnosti a lepší podporu.

Glide

V bezplatné verzi Glide lze publikovat pouze 3 aplikace na web a navíc je možné aplikaci sdílet různými způsoby, včetně URL adresy, QR kódu, nebo prostřednictvím pozvánky odeslané e-mailem. Tento přístup umožňuje uživatelům jednoduchou publikaci a rychlé sdílení aplikace. Přihlášení do aplikace je povinné a vyžaduje ověření prostřednictvím e-mailu, což zajišťuje základní úroveň zabezpečení a umožňuje sledovat a spravovat přístup k aplikaci. Když uživatel nemá v aplikaci účet na daný e-mail, tak se mu účet sám vytvoří jako nový uživatel. Tento model sdílení je ideální pro malé projekty a prototypování, kde je důležitá flexibilita a snadná dostupnost aplikace pro užší skupinu uživatelů.

Softr

Softr ve své bezplatné verzi umožňuje publikování neomezeného množství aplikací na internetu, což dává uživatelům volnost v rámci jejich vývoje a nasazení projektů. Kromě možnosti publikovat aplikace je v této verzi také poskytována možnost použít jednu vlastní doménu, což poskytuje možnost zvýšit profesionalitu a značkový vzhled aplikace. Tato funkce však nebyla využita pro aplikaci správy projektů. Sdílení aplikace je realizováno jednoduše přes URL adresu, což usnadňuje rychlé a efektivní rozšiřování mezi potenciálními uživateli. Přístup do aplikace vyžaduje přihlášení, a pokud uživatelé nemají založené účty, musí se zaregistrovat.

6.2 Porovnání nástrojů na základě připravené ukázky

Porovnání no-code nástrojů Adalo, Glide a Softr bylo provedeno na základě vytvořené ukázkové aplikace pro správu projektů. Každý z těchto nástrojů má své vlastní charakteristiky a přístupy k vývoji, které ovlivňují proces tvorby aplikací a konečný uživatelský zážitek.

Vývoj ukázkové aplikace ve všech třech nástrojích byl proveden v podobném časovém rámci, s Glide zabírajícím 21 hodin, Adalo 19 hodin a Softr 17 hodin. Je však důležité poznamenat, že vývoj v Glide trval nejdéle, protože tento nástroj byl použit jako první a musely být provedeny určité úpravy a experimenty s funkcionalitou aplikace. A v Glide byla implementována jedinečná funkce, která umožňuje automaticky vyhodnocovat, zda již byl termín splnění jednotlivých úkolů překročen. U Softr může být kratší čas způsoben tím, že nebyla nastavována viditelnost komponent, kterou bezplatná verze Softr neumožňuje.

Vývoj databáze byl jednoznačně nejsnazší v Adalo, které nabídlo intuitivní a snadné prostředí pro vytváření a správu databáze, kde byly definovány tabulky, atributy, typy dat a vazby mezi tabulkami, včetně vazby M:N. K tomu, aby uživatel zvolil správnou vazbu, Adalo poskytuje slovní popis (viz obrázek 6.1). V Glide byla databáze vytvořena v nástroji Glide Tables a pro Softr byla vytvořena databáze v externím nástroji Airtable. Obě alternativy poskytovaly podobnou možnost definování tabulek a atributů jako Adalo, ovšem Glide a Airtable umožňují širší výběr datových typů. Vazby mezi tabulkami se v těchto nástrojích vytvářely speciálními funkcemi, které nepodporovaly přímé spojení mezi tabulkami. To vedlo k nutnosti vytvoření dodatečných sloupců, které sloužily jako pohledy do jiných tabulek. Tento proces byl dosti zdlouhavý až frustrující, což by mohlo odradit potenciální uživatele. Glide navíc ani nepodporuje vazby M:N, a proto musela být vytvořena propojovací tabulka. K tomuto kroku Glide nijak uživatele nenavádí a ani neposkytuje žádného průvodce. Airtable jako Adalo podporuje tvorbu přímé vazby M:N, při vytváření

relací je vždy ukázána možnost zaškrtnout připojení k více záznamům. Když se tato možnost zaškrtně v obou tabulkách, vznikne tak relace M:N. Datové typy byly ve všech třech nástrojích uvedeny jednoduše, aby jim rozuměl i běžný uživatel bez větších znalostí datových typů.

Co se týče tvorby designu, Adalo umožňuje flexibilní návrh uživatelského rozhraní aplikace. Uživatelé mohou snadno přidávat, upravovat a propojovat komponenty, aby vytvořili přizpůsobené uživatelské zážitky. Na druhou stranu to může způsobit nekonzistenci v designu, protože jeho kvalita je z velké části závislá na autorovi, proto je potřeba, aby uživatel měl alespoň základní znalosti tvorby designu. Naopak, v Glide a Softr, které nabízejí více omezené možnosti přizpůsobení designu, je design aplikace konzistentnější a profesionálnější.

Ohledně responzivity, platforma Adalo podporuje responzivitu aplikace, přesto jsou výsledky značně neuspokojivé. Aplikaci se při změně rozlišení rozbíjí uživatelské rozhraní a aplikace je takřka nepoužitelná. U Glide, i když design obvykle funguje lépe, mohou se občas objevit problémy, jako je přerušovaný text, což může negativně ovlivnit estetiku aplikace, ale obecně zůstávají aplikace funkční. Na rozdíl od obou těchto platforem, Softr nabízí plně responzivní design, díky čemuž jsou aplikace vyvinuté v Softr plně funkční a vizuálně příjemné i na mobilních zařízeních.

Dalším důležitým faktorem je funkčnost aplikace, kde se ukázalo, že v Glide byly nutné úpravy kvůli absenci přímé podpory relací M:N a existenci propojovací tabulky. Tyto úpravy zahrnovaly zajištění integrity dat díky nastavení viditelnosti akčních tlačítek, což vedlo k omezením jako například nemožnost smazat úkol, pokud je k němu přiřazen uživatel, nebo nemožnost smazat uživatele, pokud má přiřazen úkol. Tato opatření jsou nezbytná k zabránění vzniku nekonzistence dat v propojovací tabulce a zachování integrity databáze aplikace.

Proces nasazení aplikací v Adalu byl realizován prostřednictvím sdílení URL adresy. Bezplatná verze Adala totiž neumožňuje publikaci aplikací v tradičním slova smyslu. Naopak, Glide i Softr poskytují možnost publikace aplikací. Glide umožňuje publikovat aplikace v omezeném počtu, konkrétně tři, zatímco Softr nabízí neomezené publikování a dokonce umožňuje u jedné aplikace vlastní doménu.

Každý z těchto no-code nástrojů má své vlastní výhody a omezení, které je třeba pečlivě zvážit při výběru pro konkrétní projekt. U každého nástroje je klíčové mít základní znalost modelování dat, aby bylo možné efektivně pracovat s těmito nástroji.

Adalo je vhodnou volbou pro uživatele, kteří preferují rychlý a snadný vývoj databáze a zároveň potřebují flexibilní možnosti nastavení designu. Nicméně, uživatelé musí být připraveni na to, že pokud nemají zkušenosti s designem aplikací, výsledný produkt nemusí být vzhledově příjemný pro koncového uživatele. Dále se Adalo potýká problémem s nefunkční responzivitou aplikace. A dalším omezením Adala je absence možnosti přímo publikovat vytvořenou aplikaci.

Na druhou stranu, Glide a Softr jsou ideální volbou pro uživatele, kteří jsou ochotni akceptovat určitá omezení v designu nebo nemají dostatečné dovednosti pro jeho tvorbu. Tyto platformy umožňují oproti Adalo publikování svých aplikací na webu, i když u Glide je tento počet omezen na tři. Práce s databází může být mírně složitější, hlavně u Glide, která neumožňuje přímo vytvářet vazby M:N. Responzivita vytvořených aplikací je u obou nástrojů funkční narozdíl od Adalo. U Softr ale bylo nalezeno velké omezení funkčnosti aplikace, kde bezplatná verze neposkytuje nastavení viditelnosti, což může uživatele hodně omezit, ale na druhou stranu tuto funkcionalitu nemusí vůbec potřebovat. U Softr byla během testování nalezeno další omezení, a to nemožnost smazat uživatele přímo v aplikaci. Důvodem je, že Softr si sám ukládá tabulku s uživateli a nastavení synchronizace mezi Softr a Airtable neumožňuje, aby se smazal uživatel v Airtable a automaticky se smazal i uživatel v Softr. Proto je uživatel vždy nucen kontaktovat správce aplikace, aby smazal účet danému uživateli v Softr i Airtable.

V závěru je třeba zvážit všechny tyto faktory a vybrat nástroj, který nejlépe vyhovuje potřebám a schopnostem uživatele, a zároveň je schopen poskytnout požadovanou funkcionalitu a estetický vzhled aplikace.

6.3 Srovnání UX podle Honeycomb

Hodnocení uživatelské zkušenosti (UX) se zaměřilo na komplexní přístup k analýze a hodnocení vybraných nástrojů podle modelu UX Honeycomb od Petera Morvilla. Tento model zahrnuje sedm klíčových aspektů, které budou použity k detailnímu zhodnocení toho, jak Adalo, Glide a Softr vyhovují potřebám a očekáváním jejich uživatelů. Sedmý aspekt představující hodnotnost produktu spojuje všechny ostatní aspekty do jednoho celkového výsledku. Cílem je poskytnout ucelený pohled na to, jak vybrané nástroje přinášejí hodnotu svým uživatelům.

Užitečnost

Adalo, Glide a Softr jsou nástroje, které umožňují uživatelům bez programátorských dovedností zapojit se do vývoje aplikací. Adalo je ideální pro vytváření mobilních a webových aplikací s využitím grafického rozhraní, které umožňuje flexibilní vytváření designu aplikace. Glide přeměňuje data z Google Sheets nebo podobných tabulkových nástrojů na aplikace, což je užitečné pro rychlé prototypování a automatizaci procesů, přičemž design se vytváří téměř automaticky, s možností menších úprav, což je výhodné pro uživatele s menším smyslem pro design. Softr umožňuje vytváření webových aplikací z dat v Airtable nebo Google Sheets, nabízející mnoho šablon, designových komponent a větší přizpůsobení oproti Glide. Celkově každý

z těchto nástrojů poskytuje uživatelům bez programátorských dovedností možnost efektivně a efektně se podílet na vývoji aplikací.

Použitelnost

Adalo, Glide a Softr nabízí snadno použitelné prostředí pro vývoj aplikací díky svým intuitivním rozhraním. Nicméně, při práci s daty a databázemi mohou uživatelé narazit na určité komplikace s Glide. Adalo exceluje ve snadném zvládnutí všech typů relací, včetně relací M:N, což jej odlišuje od Glide a Airtable, který byl použit u Softr, protože Softr nemá svůj nástroj na správu databáze. Airtable a Glide umožňují nastavování relací mezi tabulkami, ale systém propojení není tak plynulý jako v Adalo. Glide navíc neumožňuje přímo vytvářet relace M:N, což vyžaduje vytváření propojovací tabulky, což může být pro některé uživatele nepraktické a složité. V porovnání s těmito dvěma platformami, Adalo poskytuje nejlepší zkušenost s propojením tabulek, nabízí plynulejší a intuitivnější nastavení relací, což činí Adalo nejpoužitelnějším nástrojem pro běžné uživatele. Softr následuje s mírně složitějším procesem propojení, zatímco Glide, i přes jeho silné stránky v jiných oblastech, zůstává v tomto ohledu na třetím místě.

Žádoucnost

Adalo, Glide a Softr se zaměřují na estetiku a uživatelský design, aby přilákaly a zaujaly uživatele. Všechny tři platformy mají jednoduché a intuitivní prostředí, avšak každá z nich se liší v estetickém výrazu a přístupu k organizaci práce. Pro zvýšení uživatelského zážitku nástroje poskytují šablony pro různé aplikace s krásným designem. Adalo jich ale oproti Glide a Softr nabízí o dost méně. Všechny tři platformy kladou důraz na přitažlivost pro běžného uživatele.

Naleznutelnost

Adalo, Glide a Softr poskytují intuitivní a logicky uspořádaná uživatelská rozhraní, díky čemuž mohou uživatelé snadno a rychle nalézat potřebné funkce a nástroje. Tento přístup minimalizuje potřebu hledání a zvyšuje produktivitu, což zjednodušuje proces vývoje aplikací. Adalo se vyznačuje snadno navigovatelným menu, které urychluje učení a podporuje efektivní design. Glide se zaměřuje na rychlou orientaci v aplikaci, čímž minimalizuje uživatelskou frustraci. Softr pak poskytuje přehledné a jednoduché vývojové prostředí, což novým uživatelům umožňuje bezproblémový přístup k jeho bohaté funkcionalitě.

Přístupnost

Adalo, Softr a Glide jsou všechny platformy pro vývoj aplikací zaměřené na širokou uživatelskou základnu, ale specifické informace o jejich přístupnostních funkcích pro uživatele s postižením nejsou přímo poskytovány. Pro úplné zhodnocení jejich přístupnosti by bylo nutné prozkoumat, zda každá platforma splňuje klíčové standardy přístupnosti, jako jsou WCAG (Web Content Accessibility Guidelines). Bez těchto informací nelze s jistotou posoudit, jak dobře tyto nástroje vyhovují potřebám uživatelů s různými formami postižení.

Věrohodnost

Věrohodnost Adalo, Glide a Softr je ovlivněna jejich schopností nabídnout konzistentní a spolehlivé služby, stejně jako transparentnost a snadnost využívání jejich platebních modelů. Glide a Softr si udržují silnou reputaci díky kladným recenzím na platformách jako Trustpilot, G2 nebo Capterra, kde uživatelé chválí zejména jejich spolehlivost a uživatelskou podporu. Naproti tomu, ačkoliv Adalo je obecně hodnocen pozitivně za svou funkčnost a umožnění uživatelům snadno vytvářet aplikace, někteří uživatelé však vyjadřují negativní reakce týkající se firemní podpory, spolehlivosti systému a nekonzistence v cenových plánech Adala.

Hodnotnost

Hodnotnost Adalo, Glide a Softr je komplexní kombinací jejich užitečnosti, použitelnosti, žádoucnosti, naleznutelnosti, přístupnosti a věrohodnosti. Adalo vyniká svou schopností rychle vytvářet aplikace bez programování, což je vhodné pro malé podniky a startupy. Glide se specializuje na transformaci dat z Google Sheets do funkčních aplikací, zatímco Softr umožňuje tvorbu pokročilých webových aplikací z datových zdrojů jako Airtable. Všechny tři nástroje poskytují intuitivní uživatelské rozhraní, které usnadňuje práci bez programovacích dovedností a zvyšuje efektivitu vývoje aplikací. Estetika a uživatelský design hrají klíčovou roli v atraktivitě a použitelnosti no-code nástrojů pro tvorbu aplikací. Zde se ukazuje, že Glide a Softr přinášejí větší rozmanitost šablon a komponent než Adalo, což zvyšuje jejich přitažlivost. Co se týče použitelnosti, Adalo jasně vede díky své snadné práci s databází. Narozdíl od Airtable, který je použitý v Softr, a Glide, který je nejméně použitelný kvůli své neschopnosti přímo vytvářet vazby M:N pro uživatele. Pokud jde o naleznutelnost, Adalo, Glide a Softr nabízejí snadno procházené rozhraní, které usnadňuje navigaci a rychlé nalezení potřebných funkcí. Z hlediska přístupnosti jsou všechny tři nástroje vhodné pro širokou uživatelskou základnu, ale specifické informace o přístupnosti pro uživatele s postižením nejsou přímo poskytovány. Věrohodnost Glide

a Softr je velmi velká vzhledem k pozitivním recenzím a podpoře, zatímco Adalo někdy vykazuje nedostatky v firemní podpoře a cenové politice.

Adalo, Glide a Softr přinášejí hodnotu uživatelům bez programátorských schopností, kteří hledají efektivní způsob vývoje aplikací s důrazem na uživatelský design. Nicméně, v kontextu jejich důvěryhodnosti a žádoucnosti vynikají Glide a Softr před Adalem. Jejich lepší výběr komponent a šablon, a věrohodnost a kladné recenze uživatelů posilují jejich hodnotu a pozici jako preferovaných nástrojů pro vývoj aplikací. I tak mohou někteří uživatelé upřednostnit Adalo pro jeho snadnou použitelnost v rámci vývoje databáze.

6.4 Vyhodnocení přínosu vybraných nástrojů pro vývoj SW běžným uživatelem

V této podkapitole je vyhodnocen přínos vybraných no-code nástrojů, Adalo, Glide a Softr, pro vývoj softwaru běžnými uživateli. Přínosy těchto nástrojů jsou analyzovány na základě vývoje ukázkové aplikace pro správu projektů a srovnání jejich uživatelské zkušenosti (UX) podle modelu Honeycomb od Petera Morvilla. Cílem je sloučit obě porovnání a identifikovat klíčové faktory, které ovlivňují přijetí těchto nástrojů běžnými uživateli a jejich přínos a použitelnost pro tvorbu aplikací. Toto hodnocení nám poskytne ucelený pohled na potenciál těchto nástrojů v kontextu širší dostupnosti a demokratizace vývoje softwaru, což umožňuje široké spektrum uživatelů vytvářet aplikace s minimálními technickými dovednostmi.

6.4.1 Adalo

Adalo se ukázal jako nástroj pro rychlý vývoj aplikací bez nutnosti programování, což ho činí ideálním pro uživatele, kteří se chtějí zapojit do vývoje softwaru. Díky svému uživatelskému rozhraní usnadňuje a zrychluje tvorbu databází a návrh uživatelských rozhraní aplikací, což uživatelům umožňuje rychle přetvářet své nápady na funkční aplikace.

Nicméně, Adalo má určitá omezení, zejména v počtu dostupných šablon a komponent, které mohou být ve srovnání s konkurencí považovány za nedostačující. Například v Adalo chybí komponenta klasické tabulky, zápatí a komentářů. To může být limitující pro uživatele, kteří hledají širší možnosti přizpůsobení a nebo nechtějí moc času strávit nad designem aplikace. Dalším omezením je publikace aplikací v bezplatné verzi a nefunkčnost responzivity vytvořených aplikací, což může hodně uživatelů, kteří s aplikací chtějí pracovat na více zařízeních s různými velikostmi obrazovky.

Analýza uživatelských hodnocení na internetu ukázala, že Adalo čelí problémům v oblasti firemní podpory, systémové stability a cenové politiky, což může potenciální uživatele odradit od jeho výběru.

Přesto může být Adalo brán jako použitelný nástroj, který umožňuje běžným uživatelům efektivně se podílet na vývoji softwaru. Zejména pak pro uživatele, kteří chtějí větší volnost při vytváření designu a jednoduché vytváření databáze pro své aplikace.

6.4.2 Glide

Glide je nástroj, který se ukázal být velmi dobrý v rychlosti a jednoduchosti transformovat data do plně funkčních aplikací, což je ideální pro uživatele, kteří chtějí minimalizovat technické aspekty vývoje a soustředit se na rychlé nasazení.

I když Glide umožňuje snadné vytvoření aplikace z dat, jeho možnosti pro design jsou poměrně omezené. Uživatelé mohou narazit na to, že nejsou schopni plně přizpůsobit vzhled svých aplikací tak, aby odpovídal specifickým vizuálním požadavkům. Tato omezení však mohou přispět k větší konzistenci a profesionálnějšímu vzhledu aplikací, jelikož Glide poskytuje sady předem definovaných komponent a šablon, které zajišťují udržitelnost a vizuální soudržnost. Pokud jde o manipulaci s daty nebo vytváření databáze v Glide Tables při rozsáhlejších množství tabulek a relací, mohou být tyto procesy zdlouhavé a pro někoho bez znalostí datového modelování i dost komplexní.

Glide se ukázal být velmi prospěšným pro uživatele, kteří preferují rychlost a jednoduchost při vývoji aplikací z dat, aniž by se museli zabývat složitostmi návrhu designu. Tento nástroj je zvláště vhodný pro ty, kteří nemají rozsáhlé dovednosti v programování nebo designu. Výsledkem zhodnocení na základě tvorby správy uživatelů v nástroji Glide byl nástroj označen jako použitelný a přínosný pro uživatele s omezenými schopnostmi programování, kteří se chtějí zapojit do tvorby softwaru.

6.4.3 Softr

Stejně jako Glide, Softr se vyznačuje schopností rychlého vývoje aplikace z dat. Softr také nabízí bohatý výběr šablon a designových komponent, což uživatelům umožňuje rychle a efektivně vytvářet atraktivní webové aplikace. Přestože Softr nabízí široké možnosti pro výběr šablon a designových komponent, může být jeho flexibilita ve vlastním designu omezena ve srovnání s Adalo, které umožňuje plnější kontrolu nad designem, nebo s porovnáním s tradičním vývojem softwaru.

Při vývoji ukázkové aplikace v platformě Softr bylo nutné použít externí databázový nástroj, Airtable, protože Softr neobsahuje vlastní databázový systém. Airtable je sice na práci s databází pro běžného uživatele snazší na pochopení než Glide Tables, ale i tak může být pro běžného uživatele zdlouhavé vytvářet sloupce navíc

ve srovnání s Adalo. Navíc připojení Softr s Airtable není úplně dokonalé, protože při vytvoření komponenty výběru nebo filtru se může použít pouze sloupec označený jako primární klíč (ID). To zapříčinilo, že jména entit musely být obsaženy v ID, aby uživatel věděl, co vlastně vybírá, a neviděl jenom nic neříkající čísla. Toto omezení může trochu narušovat celkový dojem z aplikace pro koncového uživatele. Při testování byla odhalena další chyba spojená se spojením Softr a Airtable, konkrétně se jedná o nefunkčnost mazání uživatele přímo v aplikaci.

Při testování Softr pro vytvoření ukázkového příkladu správy uživatelů se ukázalo, že určitá funkcionality, specificky nastavení viditelnosti komponent, není podporována v bezplatné verzi tohoto nástroje. V důsledku toho může kdokoli v aplikaci na správu projektů upravovat nebo mazat prvky, ke kterým by standardně neměl mít přístup a navíc u aplikace nefunguje mazání uživatelů. Tato omezení značně snižují funkčnost aplikace a přínos pro vývoj softwaru ve srovnání s Adalo a Glide.

6.4.4 Vyhodnocení

Pouze Adalo a Glide se prokázaly jako velmi užitečné v rámci připravené ukázky správy projektů, i když u Adalo se ukázal problém s responzivitou vytvořené aplikace a u Glide složité vytváření vazby M:N. Softr díky svému omezení pro nastavení viditelnosti komponent a nefunkčnosti mazání uživatelů nedokázal splnit plnou funkcionality aplikace správy projektů. To ovšem neznamená, že by nástroj nebyl přínosný, někteří uživatelé se tato omezení nemusí týkat v rámci jejich aplikace.

Proto jsou nástroje Adalo a Glide vyhodnoceny jako velmi přínosné a Softr pouze z části přínosné pro běžné uživatele, kteří se chtějí podílet na procesu vývoje softwaru. Přičemž každý z nástrojů nabízí specifické výhody, které mohou zrychlit a usnadnit vývoj aplikací, snížit náklady a umožnit uživatelům bez technických znalostí efektivněji reagovat na tržní potřeby, což je v souladu se zjištěními ve studiích, které ukazují, jak no-code platformy přispívají k digitální transformaci a modernizaci softwarového vývoje [22, 23].

Hlavním cílem práce bylo porovnat nalezené nástroje pro automatizovanou tvorbu aplikací s minimální potřebou psaní programového textu podle jejich dostupnosti a poskytované funkcionality. Na základě tohoto srovnání byly vybrány tři nástroje pro podrobnější analýzu. Dále bylo provedeno porovnání použitelnosti a UX vybraných nástrojů na základě navržené ukázkové aplikace a byl vyhodnocen přínos těchto nástrojů pro vývoj softwaru běžným uživatelem. Všechny stanovené cíle byly úspěšně splněny.

Pro úspěšné dosažení stanovených cílů bylo nutné nalézt a seznámit se s nástroji, které splňují cíle práce. Tyto nástroje byly nalezeny prostřednictvím vyhledávání na internetu a pročítáním různých článků a žebříčků. Nakonec bylo vybráno pět no-code a pět low-code nástrojů, u kterých byly podrobně popsány jejich vlastnosti. Vybrané nástroje byly následně porovnány z hlediska ceny, dostupnosti a funkcionality, které poskytovaly bezplatné verze. Na základě tohoto porovnání byly vybrány tři no-code platformy: Adalo, Glide a Softr, především kvůli jednoduchosti, přístupnosti a hezkému grafickému rozhraní.

Následně byla jako ukázková aplikace navržena správa projektů. U které byla podrobně popsána její funkcionality, databáze s relacemi a use-case. Tato aplikace byla kompletně sestavena v Adalo a Glide úspěšně, a v nástroji Softr byla vytvořena bez nastavení viditelnosti komponent, protože tuto funkci poskytují pouze placené verze, a bez možnosti mazání jednotlivých uživatelů přímo v aplikaci.

Vývoj všech tří nástrojů byl podroben analýze, kde bylo odhaleno, že každý z těchto nástrojů přináší specifické vlastnosti, které mohou být významné pro určité uživatelské schopnosti a potřeby. Všechny nástroje se vyznačují velmi intuitivním uživatelským rozhraním a potřebou mít základní znalosti modelování dat. Adalo vyniká prací s databází a volností, co se týče tvorby designu, ale potýká se s problémy s responzivitou u vytvořených aplikací a špatnými recenzemi kvůli špatně navrženým cenovým plánům, firemní podpoře a nestabilitě systému. Glide oproti tomu nabízí rychlou a efektivní tvorbu aplikace z tabulkových dat, ale s velkým omezením ve volnosti designu aplikace a složitější tvorbou databáze. Softr vynikl díky široké škále designově přívětivých komponent, které pomohly zrychlit a zpříjemnit

vývoj aplikace, avšak omezení v nastavení viditelnosti komponent je významnou nevýhodou oproti Glide a Adalo. Softr dále nenabízí žádný svůj systém pro práci s daty a spoléhá se pouze na externí nástroje, jako je Airtable. U tohoto spojení byly nalezeny další problémy, a to při nastavování komponent pro filtraci a výběr, kdy bylo nezbytné upravit primární klíč (ID) jednotlivých entit. Navíc, kvůli špatné synchronizaci mezi nástroji Softr a Airtable nelze přímo v aplikaci mazat uživatele.

Výsledkem práce je, že nástroje Adalo a Glide jsou velmi přínosné pro běžného uživatele a nástroj Softr pouze částečně. Nástroje jsou přínosné hlavně z hlediska úspory času a z hlediska demokratizace vývoje funkčního softwaru, což umožňuje běžným uživatelům bez programovacích schopností zapojit se do tvorby softwaru. Nicméně, důležité je si uvědomit, že každý uživatel má odlišné potřeby a dovednosti. Proto je klíčové vybrat nástroj, který nejlépe odpovídá konkrétním požadavkům a schopnostem každého jednotlivce.

Přehled zkratk

UI User Interface

GUI Graphical User Interface

UX User Experience

PWA Progressive Web App

HTTP Hypertext Transfer Protocol

API Application Programming Interface

CRM Customer Relationship Management

NLP Natural Language Processing

GPT Generative Pre-trained Transformer

ID Identification

URL Uniform Resource Locator

WCAG Web Content Accessibility Guidelines

HTML Hypertext Markup Language

CSS Cascading Style Sheets

Bibliografie

- [1] BOCK, A. C.; FRANK, U. Low-Code Platform. *Bus Inf Syst Eng* [online]. 2021, roč. 63, s. 733–740 [cit. 2023-12-10]. Dostupné z DOI: 10.1007/s12599-021-00726-8.
- [2] *Adalo* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://www.adalo.com/>.
- [3] *Bubble* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://bubble.io/>.
- [4] *Glide* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://www.glideapps.com/>.
- [5] *AppSheet* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://about.appsheet.com/>.
- [6] *Mendix* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://www.mendix.com/>.
- [7] *Microsoft PowerApps* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://powerapps.microsoft.com/>.
- [8] *OutSystems* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://www.outsystems.com/>.
- [9] *QuickBase* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://www.quickbase.com/>.
- [10] *Softr* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://www.softr.io/>.
- [11] *Zoho Creator* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://www.zoho.com/>.
- [12] SARKAR, Advait et al. What is it like to program with artificial intelligence? *arXiv preprint arXiv:2208.06213*. 2022.
- [13] BISWAS, Som. Role of ChatGPT in Computer Programming. *Mesopotamian Journal of Computer Science* [online]. 2023, roč. 2023, s. 9–15 [cit. 2024-02-08]. Dostupné z DOI: 10.58496/MJCSC/2023/002.

- [14] YILMAZ, Ramazan; KARAOGLAN YILMAZ, Fatma Gizem. Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. *Computers in Human Behavior: Artificial Humans* [online]. 2023, roč. 1, č. 2, s. 100005 [cit. 2024-02-08]. ISSN 2949-8821. Dostupné z DOI: <https://doi.org/10.1016/j.chbah.2023.100005>.
- [15] ANI, Zhamri Che; HAMID, Zauridah Abdul; ZHAMRI, Nur Nazifa. The Recent Trends of Research on GitHub Copilot: A Systematic Review. In: *International Conference on Computing and Informatics*. Springer, 2023, s. 355–366.
- [16] NORMAN, Don; NIELSEN, Jakob. *The Definition of User Experience (UX)* [online]. Nielsen Norman Group, 1998-08-08. [cit. 2024-02-12]. Dostupné z: <https://www.nngroup.com/articles/definition-user-experience/>.
- [17] ALLAM, AH; DAHLAN, Halina Mohamed. User experience: challenges and opportunities. *Journal of Information Systems Research and Innovation*. 2013, roč. 3, č. 1, s. 28–36.
- [18] NIELSEN, Jakob; MOLICH, Rolf. Heuristic evaluation of user interfaces. In: *Proc. ACM CHI'90 Conf.* Seattle, WA, 1990, s. 249–256.
- [19] MORVILLE, Peter; SULLENGER, Paula. Ambient Findability: Libraries, Serials, and the Internet of Things. *The Serials Librarian* [online]. 2010, roč. 58, č. 1-4, s. 33–38 [cit. 2024-02-14]. Dostupné z DOI: [10.1080/03615261003622999](https://doi.org/10.1080/03615261003622999).
- [20] KRUG, Steve. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. Berkeley, CA: New Riders, 2014. ISBN 978-0-321-96551-6.
- [21] NIELSEN, Jakob. *10 Usability Heuristics for User Interface Design* [online]. Nielsen Norman Group, 1994-04-24. [cit. 2024-02-12]. Dostupné z: <https://www.nngroup.com/articles/ten-usability-heuristics/>. Revised version published 30.01.2024.
- [22] HEUSCHKEL, Simon. *The impact of no-code on digital product development*. 2023. Dostupné z arXiv: [2307.16717](https://arxiv.org/abs/2307.16717) [cs.SE].
- [23] YAN, Zhaohang. *The Impacts of Low/No-Code Development on Digital Transformation and Software Development*. 2021. Dostupné z arXiv: [2112.14073](https://arxiv.org/abs/2112.14073) [cs.CY].

Vytvořené aplikace



Příloha obsahuje odkazy na aplikace vytvořené pomocí různých no-code nástrojů. Je důležité si uvědomit, že funkčnost a dostupnost těchto aplikací v dlouhodobém horizontu nelze zaručit. Přístup k aplikacím a jejich plná funkcionální mohou být ovlivněny změnami v politice poskytovatelů těchto no-code platform, aktualizacemi softwaru, nebo externími faktory, které jsou mimo kontrolu autora práce.

U každého no-code nástroje použitého v této práci je detailně popsáno, jak se do vytvořené aplikace dostat, včetně případných přihlašovacích údajů nebo specifických instrukcí pro přístup.

Adalo

K aplikaci vytvořené v nástroji Adalo je možné přistupovat buď prostřednictvím nové registrace, nebo přihlášením se pomocí již existujícího účtu. Pro snadný přístup k aplikaci je již připraven účet, který můžete využít k testování funkcionalit bez potřeby vlastní registrace. Přihlašovací údaje pro tento účet jsou níže uvedené a umožňují okamžité přihlášení a zkoumání aplikace.

Přihlašovací údaje k připravenému účtu:

- **e-mail:** alma@example.com
- **heslo:** heslo1234

Odkaz na aplikaci:

<https://jakub-homolkas-team.adalo.com/project-manager>

QR kód pro aplikaci:



Glide

K přístupu k aplikaci vytvořené v nástroji Glide je potřeba zadat e-mailovou adresu, na kterou je následně odeslán číselný kód umožňující vstup do aplikace. Pokud na daný e-mail dosud nebyl vytvořen účet, bude automaticky vytvořen nový. Je důležité si ověřit spamovou složku vaší e-mailové schránky, protože existuje vysoká pravděpodobnost, že zpráva s přístupovým kódem může být omylem zařazena mezi spam.

Odkaz na aplikaci: <https://project-management-ar9b.glide.page>

QR kód pro aplikaci:



Softr

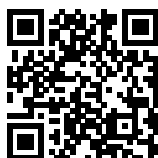
K aplikaci vytvořené v nástroji Softr je možné přistupovat buď prostřednictvím nové registrace, nebo přihlášením se pomocí již existujícího účtu. Pro snadný přístup k aplikaci je již připraven účet, který můžete využít k testování funkcionalit bez potřeby vlastní registrace. Přihlašovací údaje pro tento účet jsou níže uvedené a umožňují okamžité přihlášení a zkoumání aplikace.

Přihlašovací údaje k připravenému účtu:

- **e-mail:** alma@example.com
- **heslo:** heslo1234

Odkaz na aplikaci: <https://jeannine9530.softr.app>

QR kód pro aplikaci:



Obsah přílohy



Text_prace	Text práce ve formátu PDF spolu se zdrojovými soubory
img	Obrázky použité v textu bakalářky
manual.tex	Text práce v .tex formátu
A21B0616P.pdf	Text práce v .pdf formátu
manual.bib	Bibliografie v .bib formátu
zadani.pdf	Zadání Bakalářské práce v .pdf formátu
Readme.txt	Popis adresářové struktury

Seznam obrázků

2.1	Uživatelké rozhraní Adalo	7
2.2	Uživatelké rozhraní Bubble	9
2.3	Uživatelké rozhraní Glide	12
2.4	Uživatelké rozhraní Google AppSheet	14
2.5	Uživatelké rozhraní Mendix	15
2.6	Uživatelké rozhraní Microsoft Power Apps	18
2.7	Uživatelké rozhraní OutSystems	19
2.8	Uživatelké rozhraní Quickbase	21
2.9	Uživatelké rozhraní Softr	23
2.10	Uživatelké rozhraní Zoho Creator	25
3.1	UX Honeycomb od Petera Morvilla	35
5.1	ER diagram pro Project management	47
5.2	Use-case pro Project management	49
6.1	Popis relací v Adalo	53
6.2	Vytváření relace v Glide	54
6.3	Lookup sloupce pro vlastníka projektu v Glide	55
6.4	Vytváření relací v Airtable	57
6.5	Sloupec ID v tabulce uživatelů v Airtable	58
6.6	Nastavení seznamu projektů na domovské stránce v Adalo	60
6.7	Nastavení titulku prvku v seznamu	60
6.8	Nastavení linku pro seznam úkolů	61
6.9	Nastavení přístupových práv u editačního tlačítka	61
6.10	Nastavení kartového seznamu projektů v Glide	63
6.11	Vytváření formuláře pro přidání projektu v Glide	64
6.12	Výsledná stránka projektů v Glide	65
6.13	Nabídka bloků v Softr	66
6.14	Vytváření formuláře pro přidání projektu v Softr	67
6.15	Připojení dat k bloku v Softr	68

6.16 Stránka s detailem projektu v Softr 68

Seznam tabulek

2.1	Informace o platformě Adalo	6
2.2	Informace o platformě Bubble	8
2.3	Informace o platformě Glide	11
2.4	Informace o platformě Google AppSheet	13
2.5	Informace o platformě Mendix	14
2.6	Informace o platformě Microsoft Power Apps	16
2.7	Informace o platformě OutSystems	18
2.8	Informace o platformě Quickbase	20
2.9	Informace o platformě Softr	22
2.10	Informace o platformě Zoho Creator	24
2.11	Tabulka nalezených platforem	27
4.1	Srovnávací tabulka no-code platforem podle ceny a dostupnosti	38
4.2	Srovnání verzí zdarma no-code platforem podle funkcionality . . .	39
4.3	Srovnávací tabulka low-code platforem podle ceny a dostupnosti . . .	41
4.4	Srovnání verzí zdarma low-code platforem podle funkcionality . . .	42

101011000011100010 1100001
1010110001 10001



11010011101101001
01100001 10101
111000101011 101