

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Optimalizace využití parkovacích míst

Jakub Ďuriš

Školitel: doc. Ing. Ivan Jelínek, CSc.
Obor: Softwarové inženýrství a technologie
Květen 2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Đuriš** Jméno: **Jakub** Osobní číslo: **499245**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Optimalizace využití parkovacích míst

Název bakalářské práce anglicky:

Optimizing the use of parking spaces

Pokyny pro vypracování:

Cílem bakalářské práce je návrh a implementace aplikace pro optimální využití parkovacích míst parkoviště, kde část míst je volně přístupná a část míst je dedikována na konkrétní zaměstnance. Cílem optimalizace je dynamicky zpřístupnit aktuálně volná místa zaměstnanců, o nichž je známo, že v daný den nebudou využita zaměstnancem, aby byla k dispozici klientům firmy.

Provedte rešerši existujících přístupů k řešení, na základě této rešerše specifikujte a upřesněte požadavky na optimalizaci přidělování parkovacích míst.

Navrhněte vhodné implementační prostředky na základě rešerše těchto prostředků.

V souladu s požadavky zadavatelské společnosti vyberte optimální variantu výstupu a datové prezentace.

Navrhněte způsob testování aplikace a proveďte testování, výsledky testů vyhodnoťte.

Aplikace musí být otestována tak, aby správně pracovala i pod zátěžovým náporům uživatelů.

Výstupem bakalářské práce bude aplikace podle navržené specifikace tak, aby vyhovovala potřebám (a případným omezením) zadavatelské společnosti.

Seznam doporučené literatury:

James Ferreira, Google Script: Enterprise Application Essentials: Adding Functionality to Your Google Apps, 2012, O'Reilly Media

Maguire, M. Google Sheets Programming With Google Apps Script (2015 Revision Complete), 2015

Estes, Janelle; MacMillan, Andy User Tested: How the World's Top Companies Use Human Insight to Create Great Experiences. 2022. ISBN: 9781119844723.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Ivan Jelínek, CSc. kabinet výuky informatiky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **09.02.2023**

Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

doc. Ing. Ivan Jelínek, CSc.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Chtěl bych poděkovat panu doc. Ing. Ivanu Jelínkovi, CSc. za odborné vedení a cenné rady, které mi pomohly tuto práci zkompletovat.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Abstrakt

Cílem této práce je sestavení detailní specifikace aplikace rozvíjející digitalizaci a efektivní využití dostupných parkovacích míst firmy. Součástí je i implementace samotné aplikace v jazyce Google Apps Script a její podrobení testování.

Klíčová slova: Webová aplikace, Google Apps Script, JavaScript, parkování, optimalizace, digitalizace, projekt, specifikace

Školitel: doc. Ing. Ivan Jelínek, CSc.
ČVUT FEL,
Technická 2,
16000 Praha 6

Abstract

The goal of this work is a detailed specification of a digitization developing application enabling an efficient use of available company parking spaces. It includes the implementation of such application in the Google Apps Script programming language and subjecting it to testing.

Keywords: Web application, Google Apps Script, JavaScript, parking, optimization, digitization, project, specification

Title translation: Optimisation of Parking Spaces Use

Obsah

1 Úvod	1	8 Příloha	29
1.1 Cíl projektu	2	Bibliografie	31
1.2 Aktuální situace	2		
2 Analýza požadavků	3		
2.1 Technologická omezení	3		
2.2 FURPS+ analýza	3		
2.2.1 Funkčnost	4		
2.2.2 Použitelnost	4		
2.2.3 Spolehlivost	4		
2.2.4 Výkon	5		
2.2.5 Podpora	5		
2.2.6 Nefunkční požadavky	5		
2.3 Diagram případů užití	5		
2.4 Procesní diagram	5		
2.5 Požadavky na data aplikace	5		
2.6 Správcovský přístup	7		
2.7 Shrnutí	8		
3 Zkušební řešení	9		
3.1 Princip fungování	9		
3.2 Pozorované závěry	10		
3.3 Shrnutí	10		
4 Struktura aplikace	11		
4.1 Google Apps Script	11		
4.2 Uchování dat	12		
4.3 Struktura projektu	12		
4.4 Omezení	13		
4.5 Shrnutí	14		
5 Uživatelské rozhraní	15		
5.1 Rozdělení přístupu	15		
5.2 Hlavní obrazovky	15		
5.2.1 Rozložení pro mobilní telefon	16		
5.2.2 Rozložení pro počítač	17		
5.2.3 Ostatní obrazovky	17		
5.2.4 Správcovské zobrazení	18		
5.3 Shrnutí	19		
6 Testování	23		
6.1 Jednotkové testy	23		
6.2 Tabulkové testy	24		
6.3 Výkonnostní testy	25		
6.4 Revize kódu	25		
6.5 Testování přijetí uživatelem	25		
6.6 Vyhodnocení testování	25		
7 Závěr a shrnutí výsledků	27		
7.1 Možnosti rozvoje	28		

Obrázky

Tabulky

2.1 Diagram případů užití	6
2.2 Procesní diagram základního průchodu aplikací	6
4.1 Náhled na editor Google Script aplikací	11
5.1 Rozložení základních obrazovek na mobilním zařízení	16
5.2 Rozložení základních obrazovek na počítači	17
5.3 Obrazovky s různými druhy upozornění	18
5.4 Správcovské zobrazení umožňující úpravu údajů o parkovacích místech	19
5.5 Dialogové okno s možnostmi editace parkovacího stání	19
5.6 Nápopěda vysvětlující barevné návrhy správcovské tabulky	20
5.7 Upozornění varující uživatele před předčasným vypnutím aplikace . . .	20

Kapitola 1

Úvod

Otázka nedostatku parkovacích míst se v dnešní společnosti stává stále důležitější. Většina domácností disponuje alespoň jedním vozem, kterým jezdí na denní bázi. Z historických důvodů je v evropských zemích nicméně většina ulic ve městech velmi úzká, protože často vznikaly v době, kdy byly uzpůsobené pro pěší provoz. [1]

Od dob výlučně pěšího přesunu z místa na místo nicméně už uběhla dlouhá doba a podstatná část prostorů mezi domy je v dnešní době uzpůsobena pro provoz aut. O to méně se nabízí míst pro parkování. Z tohoto důvodu je nezbytné nakládat efektivně s místy, která jsou k dispozici. To platí o to více u velkých společností, mezi které patří i zadavatel projektu. Do práce každý den přijíždí autem stovky zaměstnanců, kteří potřebují zaparkovat. Počet míst je ovšem omezen. Je proto potřeba co nejlépe všechna dostupná místa zužítkovat.

Právě z toho důvodu je cílem tohoto projektu vytvoření aplikace umožňující efektivní využití a digitalizaci parkování před budovou zadavatelské firmy. Tedy zajištění zužítkování i těch parkovacích stání, která by za běžných okolností zůstala prázdná.

Parkoviště je důležitou, ale často zanedbávanou možností firmy, jak udělat dobrý první dojem i na návštěvníky. To platí obzvláště v případě, že se jedná o parkoviště firmy zabývající se vývojem komponentů do aut, které zahrnují vše od elektrických součástek až po systémy autonomního řízení. Služba, jež pomůže každému zájemci najít dostupné parkovací místo, může v očích návštěvníka zvednout mínění o dané společnosti ještě před vstupem do budovy.

Elektronická správa parkoviště usnadňuje profesní život hlavně řadovým zaměstnancům. Ti díky ní získají schopnost najít místo k zaparkování rychleji a jednodušeji. Tím se pro ně zkracuje čas tím strávený a spolu s ním i celková doba transportu do práce a z ní. Zároveň představuje možnost, jak získat přehled o stavu kapacity parkoviště ještě před samotným příjezdem. Tento projekt se proto zabývá návrhem a vývojem aplikace zaměřené na využití těmito zaměstnanci, kteří z jejího zavedení do provozu budou v každodenním životě nejvíce profitovat.

1.1 Cíl projektu

Cílem projektu je aplikace pro optimální využití parkovacích míst parkoviště, kde část míst je volně přístupná a část míst je určena pro konkrétní zaměstnance. Cílem optimalizace je dynamicky zpřístupnit aktuálně volná místa zaměstnanců, o nichž je známo, že v daný den nebudou využita, jiným zájemcům z řad společnosti

Součástí projektu je i splnění požadavků zadavatelské společnosti na podobu a funkčnost aplikace. Ty vyplynou z analytické části projektu, kde bude rozebráno současné řešení a stav a bude vytvořena specifikace pro aplikaci.

Celý systém je pro firmu prvním krokem k automatizaci a digitalizaci procesu parkování a má poskytnout možnost v budoucnu snadno rozšířit svou funkcionalitu. Elektronická správa míst totiž umožňuje implementaci dalších nadstaveb, jakou je například aktuální zobrazení obsazenosti parkovacích stání. Tuto informaci je následně možné zajímavě prezentovat, ať už na interaktivním kiosku u vjezdu, v online aplikaci umožňující rezervaci konkrétního místa, nebo jednoduše na obrazovce informující o počtu volných stání.

1.2 Aktuální situace

Parkoviště společnosti disponuje kapacitou několika desítek parkovacích míst. Většina těchto míst není nikomu přiřazena, stát na nich smí všichni zaměstnanci. Zbytek parkovacích stání je dedikovaný konkrétnímu uživateli, nebo pro konkrétní účel. Může například být vyhrazené pro konkrétního zaměstnance ve vyšším managementu, případně pro parkování sdílených firemních vozů.

Všechna parkovací místa mají zaznamenaný aktuální způsob využití, tedy v případě, že místo slouží jako dedikované, je zřejmé, komu bylo přiřazeno. Tento zaměstnanec má následně výhradní právo k využívání místa, nehledě na skutečnost, jestli se zrovna nachází v kanceláři, potažmo jestli ho daný den využije. To znamená, že v případě, kdy parkování neobsadí, bude místo celý den prázdné, jelikož nikdo jiný k němu nemá přístup. Tento způsob využití dostupných parkovacích míst je velmi neefektivní, neboť ostatní parkovací místa se poměrně rychle zaplní a další zaměstnanci již nemají kde zaparkovat, i přesto, že spousta parkovacích stání zeje prázdnotou.

Kapitola 2

Analýza požadavků

2.1 Technologická omezení

Ve firmě probíhá běžná práce v aplikacích od společnosti Google – Sheets, Docs, Drive. Všechny automatické systémy se spouští nad tabulkou Google Sheets a jsou psány ve skriptovacím jazyce na bázi JavaScriptu, Google Apps Script. Ten využívá tabulkové funkce dostupné v Sheets a rozšiřuje je mimo jiné o webové aplikace s vlastním uživatelským prostředím[3]. Z tohoto důvodu je nutné celou aplikaci implementovat v tomto programovacím jazyce a nad jednou konkrétní tabulkou Google Sheet.

Další potíž nastává s databází, která není ve firmě vůbec využívána. Data se ukládají do tabulek Google Sheets a stejný postup bude muset být dodržen i zde. Tabulky by se snad daly nazvat jakýmsi prostým databázovým souborem[4], nicméně oproti klasickým SQL databázím mají spoustu nevýhod. Mezi hlavní patří složitější přístup ke konkrétním datům z důvodu nutnosti simulace SQL dotazů jinými způsoby. Konkrétně jsou data z tabulek získávána pomocí funkcí ze třídy Sheet, což je interní třída Google Script aplikací. Hodnoty z listu konkrétní tabulky jsou uloženy do dvourozměrného pole, také nazývaného matice[5]. Zde vznikají hned dva problémy. Prvním je samotné načítání těchto hodnot, které je velmi pomalé, neboť vyžaduje otevření tabulky, načtení dat a jejich návrat. Při přístupu k těmto datům je tedy z důvodu zachování rozumné rychlosti běhu programu nutné načítat všechna data naráz, pracovat s nimi, a potom opět všechna aktualizovat. To je náročné na paměť.

Po načtení informací přichází druhý zádrhel, který je způsoben tím, že hodnoty jsou uloženy v matici, což si vynucuje dohledávání konkrétních hodnot procházením celého pole. To opět zpomaluje běh programu.

2.2 FURPS+ analýza

FURPS analýza je nástroj, který analyzuje a prioritizuje požadavky (na systém). Skládá se z 5 částí: [6]

- Funkčnost (Functionality)

- Použitelnost (Usability)
- Spolehlivost (Reliability)
- Výkon (Performance)
- Podpora (Serviceability)

Znaménko + v tomto případě znamená rozšíření o nefunkční požadavky. Ty nepopisují co bude systém dělat, ale jak to bude dělat[7]. Například upřesňují barevné rozložení systému, nebo v jakých jazycích má být dostupné uživatelské prostředí.

2.2.1 Funkčnost

V aplikaci bude možné zobrazit si aktuálně dostupná dedikovaná parkovací místa a zarezervovat si je pro vlastní použití. Přístup zaměstnanců k rezervaci bude nicméně omezen a bude se manuálně spouštět ve chvíli, kdy se ostatní parkovací místa zaplní. Nebude zároveň možné rezervovat si všechna dostupná místa na parkovišti, do systému budou vložena pouze dedikovaná parkovací stání, která jsou většinou přiřazena zaměstnancům managementu.

Aplikace bude obsahovat také rozhraní pro správu parkovacích míst, kterou budou moci využít zaměstnanci se správcovskými právy. V té bude možné konkrétní místo odstranit z nabídky, aby nebylo možné vytvořit si na něm rezervaci ani během nepřítomnosti jeho majitele. Dále bude možné místa přidat nebo odebrat z nabídky pouze pro daný den. Taktéž se odsud bude celá aplikace spouštět.

2.2.2 Použitelnost

Webové rozhraní bude optimalizováno pro prohlížeče Google Chrome a Microsoft Edge. Očekává se primárně responzivní návrh umožňující jednoduché použití z mobilních telefonů s operačním systémem Android. To z důvodu, že většina uživatelů bude do aplikace přistupovat během cesty do práce, aplikace je proto nesmí nutit věnovat jí více pozornosti, než je nezbytně nutné.

2.2.3 Spolehlivost

Žádné konkrétní požadavky na spolehlivost nebyly vzneseny, předpokládá se tedy běžně přijatelná dostupnost dat. Přesnost vrácených dat je závislá výlučně na vstupních informacích, které nejsou ve správě této aplikace. Zásadní je co nejvyšší spolehlivost v ranních hodinách, kdy lze předpokládat nejvyšší nápor uživatelů. Informace o vytvořených rezervacích musí být správné za každé situace.

Aplikace nesmí obsahovat žádné kritické chyby, které by znemožňovaly její použití. Všechny tyto aspekty budou ověřeny testovací fází, ve které proběhne zejména kontrola správnosti běhu páteřních funkcí a robustnost při větším náporu uživatelů.

■ 2.2.4 Výkon

Očekává se běžně uživatelsky akceptovaný výkon a rychlost načítání dat, v nejvytíženější moment může být aplikace naráz využívána více než sto zaměstnanci. Datový výstup by měl být zobrazen do 5 vteřin.

Zároveň je ale nutné ověřovat správnost dat v každém kroku vytváření rezervace uživatelem. Konkrétně to znamená, že každá akce ze strany uživatele musí být překontrolována aktualizací vstupních dat z tabulky, což bude běh poměrně zdržovat.

■ 2.2.5 Podpora

Kompletní dokumentace systému bude ve vlastnictví zadavatele, ten bude zároveň zajišťovat udržování a podporu aplikace.

■ 2.2.6 Nefunkční požadavky

Uživatelské rozhraní bude dostupné v anglickém jazyce a bude navrženo ve stylu co nejpodobnějším ostatním aplikacím zadavatele. To se týká především fontu písma a barevného návrhu. Všechny stránky a části aplikace musí také obsahovat logo společnosti.

■ 2.3 Diagram případů užití

Diagram případů užití, anglicky Use Case diagram, viz obrázek 2.1, ukazuje základní funkcionality systému pro všechny zúčastněné strany. Z nich je zásadní zaměstnanec, pro kterého je celá aplikace určena. Ten ji využívá pro rezervaci parkovacího místa, na kterém následně může bezstarostně zaparkovat.

Významnou součástí aplikace je i správcovská konzole. V ní může správce systému spravovat všechna dedikovaná místa. Mezi její základní prvky patří možnost odebrat místo ze seznamu nabízených.

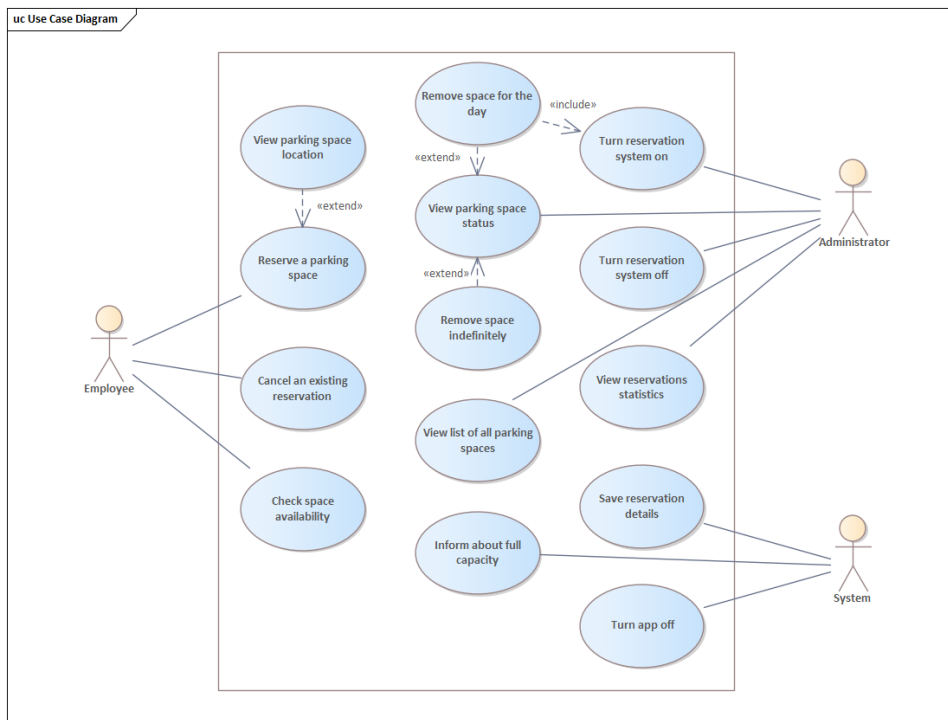
■ 2.4 Procesní diagram

Proces na obrázku 2.2 zobrazuje základní průchod aplikací ve chvíli, kdy se naplní kapacita zbytku parkoviště. Správce systému v tu chvíli webové prostředí spustí a zaměstnanci mohou začít žádat o parkovací místa.

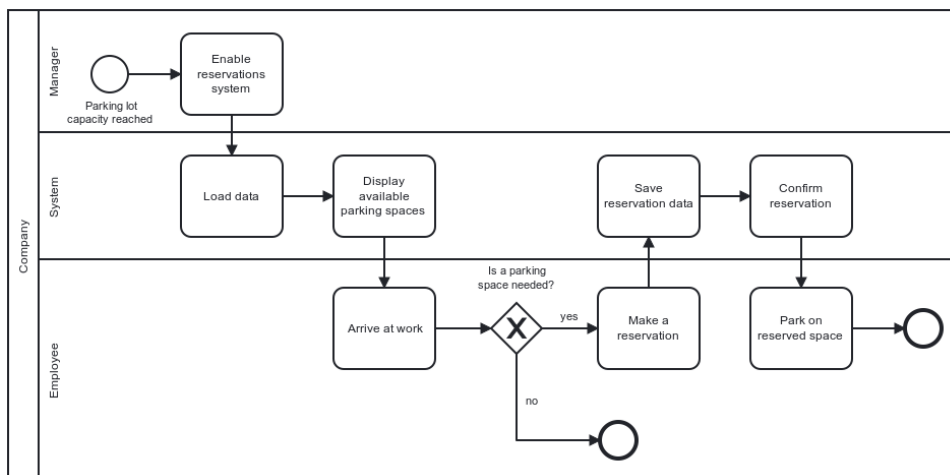
■ 2.5 Požadavky na data aplikace

Systém by měl zaměstnancům umožnit zarezervovat si jedno z dostupných dedikovaných míst. Seznam dostupných míst s identifikačními čísly jejich majitelů a schválené absence pro daný den jsou ve správě jiných oddělení firmy, tedy požadavky na kvalitu těchto vstupních dat nejsou specifikovány, ale

2. Analýza požadavků



Obrázek 2.1: Diagram případů užití



Obrázek 2.2: Procesní diagram základního průchodu aplikací

aplikace z nich má sama vycházet. Vše je uloženo v Google Sheets. Všechna data vystupující z aplikace musí být opět uložena do tabulky Google Sheets.

Dostupné informace se nachází ve 2 různých tabulkách. První tabulkou je prostý seznam identifikačních čísel všech zaměstnanců, kteří mají na daný den schválenou absenci v docházkovém systému. Tabulka je automaticky generovaná a aktualizuje se každý den v ranních hodinách. Druhá z tabulek je rozsáhlejší a obsahuje všechny údaje o parkovištích a místech. Je ve správě personálního oddělení.

Pro účely systému je nutné všechny listy se zdroji kopírovat do třetí tabulky, která může být na rozdíl od originálů editována. Nad tou poběží celá aplikace. V této tabulce se vytvoří seznam všech osob ze seznamu absencí, které mají zároveň přiřazené parkovací místo. Tato stání se potom budou nabízet ostatním zájemcům k zaparkování. Musí být dále možné tento seznam manuálně upravit tak, aby bylo možné konkrétní místo z něj vyndat, nebo do něj naopak vložit. To přijde vhod, pokud vzniknou nesrovnalosti v přehledu absencí. Současně bude tabulka sloužit pro ukládání informací o systému. Bude tedy obsahovat mimo jiné stav jeho zapnutí, odkazy na obrázky nebo seznam emailových adres zaměstnanců se správcovským přístupem. Důležitým aspektem je, že aplikace nebude ukládat žádné časové údaje. Její běh bude ohraničen manuálním zapnutím ráno a automatickým vypnutím na konci dne. Bude totiž koncipována jako jakýsi nouzový způsob, jak vytvořit na parkovišti ještě o trochu více míst k zaparkování. Není totiž žádoucí, aby si kdokoliv mohl zarezervovat konkrétní místo na několik dní dopředu, a to hned z několika důvodů. Prvním a nejdůležitějším je fakt, že dedikovaná místa jsou stále platným bonusem k manažerské pozici. Vzniká proto riziko následující situace. Manažer má schválenou několikadenní absenci. To někomu jinému umožní rezervovat si jeho místo na několik dní. Tento zaměstnanec už si pravděpodobně nebude každý den kontrolovat aplikaci, jestli má místo stále zarezervované. I kdyby bylo řečeno, že se toto kontrolovat musí, může na to zapomenout, nebo může auto nechat v práci přes noc. Do toho se manažerovi změní rozvrh, najednou bude nečekaně potřebovat do kanceláře a při příjezdu zjistí, že je jeho místo obsazeno. Omezení vytváření rezervací pouze na jeden daný den toto riziko zmírňuje. Dalším argumentem proti několikadenním rezervacím je situace, kdy si zaměstnanec zablokuje parkování dopředu a pak ho nevyužije. Například dorazí do práce veřejnou dopravou. Tím je opět místo zabráno, ale nevyužito.

2.6 Správcovský přístup

Kromě rezervací parkovacích míst bude aplikace umožňovat i jejich správu. Přístup do této části bude umožněn vybraným zaměstnancům z personálního oddělení, kteří měli i doteď parkování na starost. Tito správci se mohou libovolně měnit. Proto musí pověřená osoba mít možnost upravit jejich seznam v aplikaci.

Osoba s přístupem do správcovského režimu bude mít možnost vstoupit do přehledu parkovacích míst, kde bude možné provést základní úpravy. Zejména

pak takové, které budou mít přímý vliv na chod aplikace. Systém bude nicméně koncipován jako jednorázová možnost, jak vytvořit na parkovišti pár míst k zaparkování navíc. Proto ani správcovská konzole nebude poskytovat příliš rozšířenou funkcionalitu, přesto by ale měla umožnit následující úkony:

- Zapnutí a vypnutí aplikace
- Odebrání místa z nabídky natrvalo
- Odebrání místa z nabídky pro daný den
- Základní statistické údaje

2.7 Shrnutí

Výstupem bude webová aplikace vytvořená nad tabulkou Google Sheets. Ta zaměstnancům firmy umožní jednoduchou rezervaci volného místa k zaparkování. Z vygenerovaného seznamu schválených absencí pro daný den se vyberou ti zaměstnanci, kteří mají přiřazeno dedikované parkovací stání, a ta se nabídnou ostatním zájemcům z řad společnosti.

Na aplikaci nebyly vzneseny žádné speciální požadavky ohledně rychlosti a dostupnosti dat. Očekává se proto dostatečný výkon, který nebude uživatele zásadním způsobem omezovat. Načtení dat by mělo proběhnout do 5 vteřin. Nejdůležitějším aspektem je poskytnutí dostatečně aktualizovaných a správných záznamů ze strany ostatních oddělení firmy. Dále musí být snadno použitelná na mobilním zařízení. Největší nápor uživatelů se očekává v ranních hodinách, hned po spuštění.

Hlavním uživatelským procesem je přihlášení zaměstnance k využívání daného parkovacího stání konkrétní den. Přístup k tomuto rezervačnímu systému bude spuštěn manuálně jeho správcem ve chvíli, kdy se kapacita volně dostupných míst zaplní.

Kapitola 3

Zkušební řešení

Přechod na řešení, které umožňuje i řadovým zaměstnancům parkovat na manažerských místech, která jsou sama o sobě bonusem k vyšší pozici ve firmě, může být pro některé zaměstnance nepříjemný. Doteď platilo, že jim je jejich stání přístupné kdykoliv a na neomezenou dobu. Zároveň je nutné zajistit, aby zaměstnanci přistupovali k parkování na manažerských místech jako k nouzové výhodě a podle toho se chovali. Znamená to totiž, že v případě vytvoření rezervace bude nutné parkovat na správném místě a včas toto stání opět uvolnit. Tím se mitiguje riziko, že manažer nebude moci na svém místě zaparkovat, i když to bude potřebovat. Aby se při spuštění plánovaného systému předešlo nedorozuměním, byla vytvořena jednoduchá aplikace, která přiřazování manažerských stání zaměstnancům otestuje v praxi.

3.1 Princip fungování

Základní logika testovací aplikace je jednoduchá. Z dostupných zdrojů se vytvoří seznam volných parkovacích míst na daný den a ten se vygeneruje v tabulce, kterou je možné vytisknout. Vytisknutý seznam míst dostane na starost pověřená osoba, v tomto případě vrátný, a ta má možnost parkovací stání po naplnění kapacity parkoviště nabízet příjezdějícím zaměstnancům. Ti potom na místě rovnou zaparkují a jejich jméno si pověřená osoba zapíše do zmíněného seznamu, pro případ nouze.

Toto triviální řešení naučí zaměstnance, že nyní existuje možnost dodatečného parkování v případě naplnění kapacity parkoviště a manažery, že jejich místa mohou být v době jejich nepřítomnosti využita někým jiným. Zároveň je však správa přiřazování stání zaměstnancům ponechána člověku, který se fyzicky nachází na parkovišti a může flexibilně vygenerovaný seznam upravovat. Například vidí, že místo, které by mělo být prázdné, je přesto obsazeno, a může ho ze seznamu vyjmout. Navíc hlídá zaměstnance, aby parkovali na příslušných místech. Na druhou stranu se zde přidává práce jednomu zaměstnanci, který nemusí být ochotný přijíždějícím kolegům místa nabízet. Proto je vhodné do budoucna nahradit přiřazování míst automatickým systémem.

3.2 Pozorované závěry

Běh testovací aplikace ukázal, že data sbíraná z docházkového systému jsou poměrně přesná, tedy že místa, která mají být prázdná, většinou skutečně prázdná jsou. Stále ale vznikaly v některých situacích problémy, že manažer s parkovacím místem přijel do práce, i když neměl. V zájmu plynulého chodu aplikace je nutné, aby se všichni zaměstnanci s novým chodem věci ztotožnili, což znamená hlavně správné vyplňování docházkového systému manažery.

Pozorování nicméně ukázalo, že je obtížné zrušit již vytvořenou “rezervaci.” Tedy ve chvíli, kdy běžný zaměstnanec zaparkuje na propůjčeném stání a začne pracovat, už není jednoduché ho kontaktovat. K vytvořeným rezervacím se tedy bude přistupovat jako k finálním, nicméně kontaktní údaje o rezervující osobě budou i tak stále dostupné, pro případ nouze.

3.3 Shrnutí

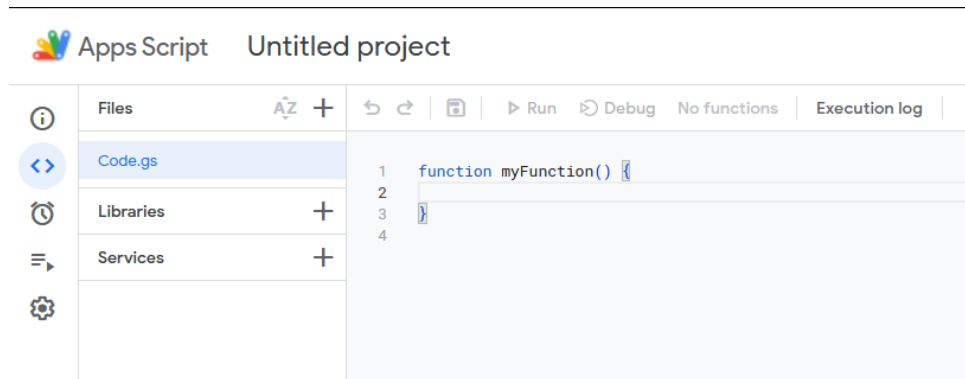
Provoz zkušebního systému ukázal, že s dostupnými daty žádné zásadní problémy nejsou, ale že je nutná určitá zvykací fáze pro všechny uživatele rezervační aplikace. Zaměstnanci se musí chovat zodpovědně a neomezovat vlastníky míst nad dohodnutý rámec. Manažeri s přidělenými místy musí poctivě a přesně vyplňovat docházkový systém tak, aby se sami o místo nepřipravili a zároveň aby jej uvolnili ostatním v době osobní nepřítomnosti.

Kapitola 4

Struktura aplikace

4.1 Google Apps Script

Google Apps Script, zkráceně GAS, je platforma pro tvorbu kódu a vývoj aplikací, zakomponovaná do prostředí Google Apps. Umožňuje přidání funkcionalit do tabulek, Gmailu, Sites a dalších služeb společnosti Google. Skripty mohou být vytvořeny jako samostatné soubory v úložišti Drive, uvnitř dokumentu nebo tabulky, případně jako Google Site[8]. GAS poskytuje možnost snadno a rychle vytvářet firemní aplikace, které se snadno propojí s Google Workspace. Vývoj probíhá v jazyce JavaScript, v editoru dostupném přímo v prohlížeči, viz. obrázek 4.1. Ten poskytuje přístup ke knihovnám umožňujícím práci s většinou aplikací[9].



Obrázek 4.1: Náhled na editor Google Script aplikací

V rámci projektu představuje Google Script vhodnou platformu pro aplikaci, protože zdrojová data jsou uložena v tabulkách Sheets. Nejvhodnější podobou je webová aplikace postavená nad jednou z těchto tabulek, která kopíruje data uložená v ostatních. Přizpůsobení aplikace pro použití ve webovém prohlížeči poskytuje několik výhod. První a nejdůležitější je uživatelský komfort. Díky kaskádovým stylům, tedy CSS, je možné nastylovat aplikaci tak, aby obsahovala velká tlačítka, která jsou snadno na dosah ruky při přístupu z mobilního telefonu. To by nebylo možné, pokud by se rezervace vytvářely například ručně přímo v tabulce. Druhou výhodou webu je právě

zmíněné odstínění uživatele od tabulky, díky čemuž není nutné, aby k ní měli přístup všichni potenciální uživatelé aplikace. V neposlední řadě je použití přes webovou stránku velmi jednoduché a pro uživatele intuitivní. Už jen proto, že není nutné stahovat žádné další programy.

4.2 Uchování dat

Aplikace vytvořené v GAS jsou navrženy tak, aby se daly využít zdroje uložené mimo jiné v tabulkách Google Sheets. Tím pádem není nutné v těchto skriptech využívat klasických databází. To nicméně přináší určitá omezení, která plynou ze složitějšího přístupu k datům. V základní podobě jsou hodnoty z tabulek získávány ve formátu dvourozměrných listů, kde jednotlivé záznamy prvního listu obsahují data z řádků, pozice (index) v každém z těchto řádků představuje sloupec [10]. Analogicky probíhá i uchovávání dat. Po zvolení konkrétního rozsahu buněk v tabulce, do níž chceme zapsat, jim lze přiřadit hodnoty z dvourozměrného listu.

Výhodou tohoto přístupu je jednoduchost implementace v aplikacích o menších rozměrech. S daty je totiž možné pracovat jako s běžným listem, který se následně snadno uloží. Problém nastává ve chvíli, kdy je nutné najít v tabulce konkrétní hodnotu nebo hodnoty. Protože nelze zadat vyhledávací parametry už během přístupu k uloženým informacím, je i z hlediska rychlosti nutné pracovat s celým rozsahem dat, ve kterém se následně vyhledává postupným procházením všech hodnot.

4.3 Struktura projektu

Obecně je webová aplikace v GAS v základu podobná klasickému webovému vývoji, má však svoje omezení. Základním rozdílem je, že je možné pracovat pouze se soubory typu .gs a .html. Soubory typu .gs obsahují funkce, které jsou vykonávány na serverové straně, tedy se pomocí těchto funkcí například přistupuje k datům z tabulek. Soubory .html obsahují zdrojový kód pro zobrazení webových stránek. K datům ze serveru se přistupuje speciálními metodami dostupnými přes JavaScript běžící na klientské straně. Ty jsou obsaženy v knihovnách přímo od Google. Skript může být publikován ve formě webové aplikace, pokud splní následující podmínky:[11]

- Obsahuje doGet(e) nebo doPost(e) funkci
- Ta vrací buď HTML service HtmlOutput objekt nebo Content service TextOutput objekt

HTML i Content service jsou příklady knihoven Google, které umožňují propojení klientské a serverové strany v Google Apps skriptu. HtmlOutput i TextOutput objekty obsahují kód ve formátu .html, který má být následně vyrenderován v prohlížeči. Je proto možné kontrolovat zadané podmínky v rámci funkce doGet() a podle jejich splnění či nesplnění zobrazovat konkrétní stránku. Obě metody se volají automaticky při přístupu na stránku.

V rámci tohoto projektu byl vytvořen jeden soubor `Code.gs`, který obsahuje všechny funkce potřebné pro přístup k datům a k jejich ukládání. Pro přehlednost a snadnou údržbu je využíván soubor `Constants.gs`, ve kterém jsou zaznamenány všechny důležité konstanty. Mezi ně patří identifikační kódy jednotlivých tabulek, názvy listů, sloupců a “souřadnice” konkrétních dat v příslušných tabulkách. Pro zajištění této možnosti je využito základní funkcionality editoru, která způsobuje, že všechny `.gs` soubory jsou mezi sebou propojené a mohou navzájem využívat metody a proměnné z ostatních.

Pro udržení základní struktury souborů webové aplikace má každá použitá “stránka” vlastní soubor ve formátu `html`. Aby se tyto soubory zmenšily, jsou všechna stylovací pravidla (která by běžně měla vlastní soubor nebo byla zapsána v HTML tagu “`<style>`”) pro každou stránku uložena v separátním souboru. Ten se jmenuje stejně jako odpovídající stránka, ale má přídomek “`CSS`.” Jeho obsah se sloučí se zbytkem během volání funkce `doGet()` a to pomocí funkce `include()`.

4.4 Omezení

GAS s sebou přináší určitá omezení, která se v běžné praxi nevyskytují. Příkladem je už fakt, že v souborové struktuře je možné pracovat pouze se soubory typu `.gs` a `.html`. To znemožňuje mimo jiné přidávání obrázků a dalších zdrojů přímo do struktury aplikace. Obrázky musí tudíž být nahrány přes veřejný odkaz, na konkrétním příkladu zde popisovaného systému to znamená práci navíc při zobrazování konkrétních obrázků se zvýrazněnou pozicí parkovacího místa. Každý obrázek je totiž nutné nahrát například na Google Drive, sdílet, a jeho ID vložit do odkazu, který obrázek na stránce správně zobrazí. Při zobrazování mapy ke konkrétnímu místu potom tedy nestačí zobrazit třeba soubor “`parkingspace12.jpg`,” kde se dá číslo snadno nahradit jiným, ale je nutné uložit si všechny odkazy a zpětně je dohledávat.

Další překážka vzniká při snaze přidat metadata do hlavičky `.html` stránky. Klasický postup, kde se jednotlivé značky jako například “`<title>`” nebo “`<meta>`” přidají do elementu “`<head>`,” zde selhává. Tyto informace je nutné přidat jako atributy k objektu `HtmlService`, který vrací funkce `doGet()`, jak je popsáno v sekci 4.3. Z běžného

```
<html>
  <head>
    <title>Example title</title>
    <meta name="viewport" content="width=device-width,
      initial-scale=1, minimum-scale=1" />
  </head>
</html>
```

tedy vznikne:

```
HtmlService.createTemplateFromFile("File").evaluate()
.setTitle("Example title")
.addMetaTag('viewport', 'width=device-width, initial-scale=1');
```

Poměrně zásadním problémem je fakt, že aplikace vytvořené v GAS neumožňují top-level navigaci nevyžádanou napřímo uživatelem. To v praxi znamená, že jakékoliv přesměrování může proběhnout jen a pouze po přímé interakci uživatele se stránkou. V kombinaci s vynucováním výlučně top-level navigace vzniká problém, že není možné automaticky znovu načíst stránku po dokončení běhu jiné funkce. V našem případě tedy není možné automaticky uživatele přesměrovat na stránku s potvrzením po úspěšném vytvoření rezervace parkovacího místa. Jedinou možností je nahrazení celé stránky novou pomocí klientského JavaScriptu. Tak je možné zajistit, že se nejprve ověří, zda-li rezervace byla vytvořena, a teprve potom je uživateli zobrazena stránka s potvrzením.

4.5 Shrnutí

Google Apps Script je platforma pro tvorbu business aplikací od společnosti Google, která pomocí knihoven umožňuje práci s ostatními aplikacemi, jako je Google Sheets nebo Google Docs. Zde je tato platforma využita pro vytvoření webové aplikace pracující s daty uloženými v tabulkách Sheets. Soubory typu .gs (Google Script) fungují na bázi skriptovacího jazyka JavaScript a představují serverovou stranu aplikace. Pomocí nich se mimo jiné přistupuje k datům uloženým v tabulkách. Ta jsou potom zobrazena na stránkách pomocí .html souborů.

Kapitola 5

Uživatelské rozhraní

5.1 Rozdělení přístupu

Webová aplikace vytvořená v rámci tohoto projektu je rozdělena na dva typy uživatelů. Výchozí práva mají všichni uživatelé, kteří mají do aplikace přístup. Ti si mohou pouze vytvořit rezervaci, zobrazit číslo přiřazeného místa, jeho polohu na plánu parkoviště, případně mohou rezervaci také zrušit. Systém musí být pro tuto skupinu co nejjednodušeji přístupný, proto je omezen pouze na základní tlačítka a jasně srozumitelný text. Pro snadnou navigaci po stránkách jsou použity velké fonty, široké mezery mezi prvky a rozměrná navigační tlačítka.

Druhou skupinou uživatelů jsou správci. Správcem může být jakýkoliv zaměstnanec, jehož email je uveden v seznamu správců. Primárně se předpokládá, že jimi budou zaměstnanci personálního oddělení, kteří i dosud měli parkoviště a parkovací místa na starosti. Tito uživatelé mají základní funkcionalitu rozšířenou o možnost vstupu do správcovské konzole. To jim ale neodebírá schopnost vytvořit si vlastní rezervaci na parkování, tak jako běžný uživatel. Správcovský přístup obsahuje 3 hlavní prvky:

- Ovládání aplikace - zapnutí a vypnutí
- Základní přehled obsazenosti míst
- Seznam všech manažerských parkovacích stání

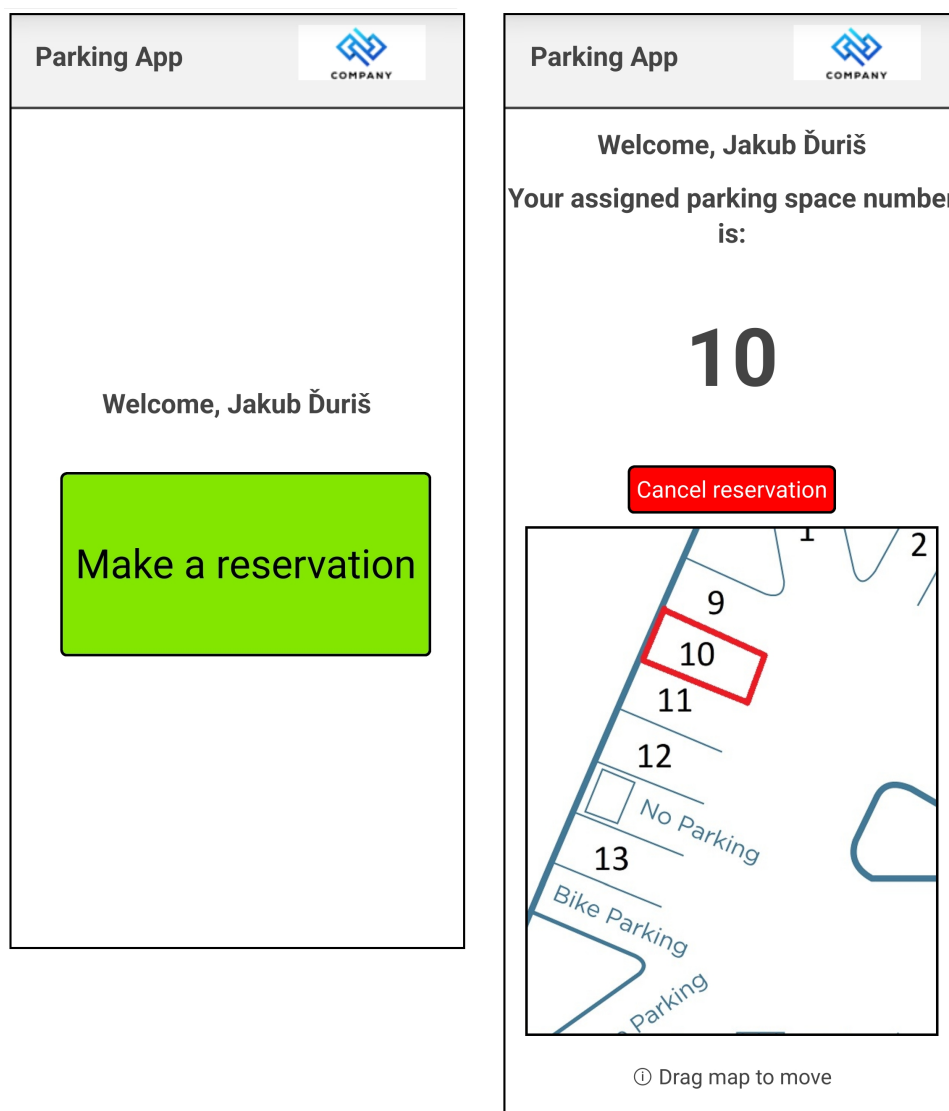
5.2 Hlavní obrazovky

Pro běžného uživatele se za hlavní rozložení považuje mobilní, protože se předpokládá nejčastější přístup na stránku právě z telefonu. Správcovská konzole je zaměřena na počítačové rozložení, protože obsahuje velkou tabulku, která se pohodlněji spravuje právě na něm.

Samozřejmostí v každém rozložení jsou hlášky zpětné vazby, informující uživatele o tom, že interakce byla zaregistrována a probíhá zpracování dat. Při přístupu na každou stránku se nejprve zobrazí načítání, dokud nejsou všechny informace načteny. Správci mají v záhlaví každé stránky možnost

přepínat mezi běžným a správcovským zobrazením. Běžným uživatelům se toto tlačítko nezobrazuje.

5.2.1 Rozložení pro mobilní telefon

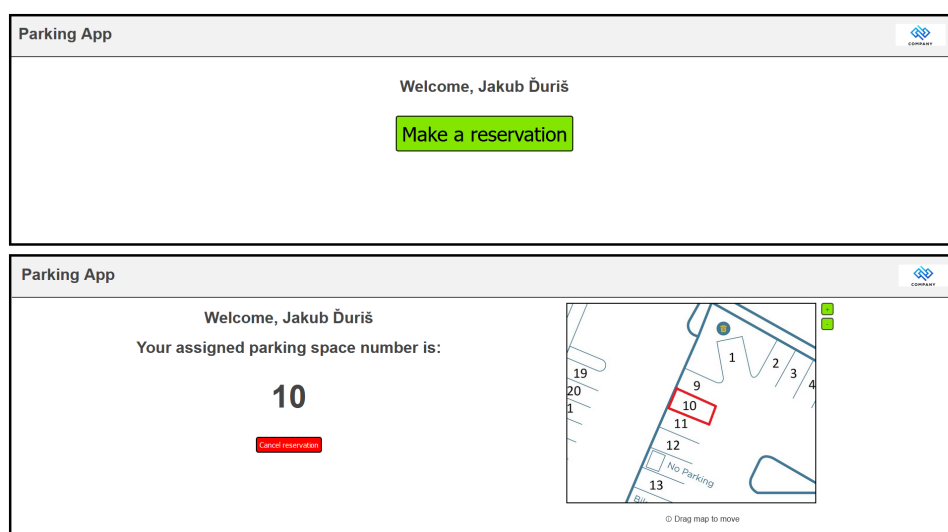


Obrázek 5.1: Rozložení základních obrazovek na mobilním zařízení

Rozložení pro mobilní telefon je koncipováno tak, aby bylo co nejjednodušší ke čtení i k ovládání. Hlavní stránka (viz. obrázek 5.1 vlevo) obsahuje velké tlačítko, které je na stránce navíc posunuto níže, aby se na něj dalo snadněji dosáhnout v případě, že je telefon držen v jedné ruce. Po jeho stisknutí je vytvořena rezervace a uživatel je přesměrován na stránku s potvrzením (viz. obrázek 5.1 vpravo). Tam je uvítán pozdravem se svým jménem, které má pasivně poskytovat zpětnou vazbu, že je vše v pořádku. Jméno uživatele se získává přes knihovnu přistupující k jeho seznamu kontaktů, ve kterém je

uložen jeho vlastní kontakt se jménem. Mimo jiné může sloužit jako kontrola, že je přihlášen správným účtem. Pod jménem se zobrazuje číslo systémem přiděleného parkovacího stání, tlačítko pro zrušení rezervace a obrázek s vyznačeným místem na plánu parkoviště. Parkovací místa jsou seřazena podle vzdálenosti od vchodu, to znamená, že nejprve jsou zarezervována místa nejbližší ke vchodu do budovy a pak postupně dále. Mapku je možné tažením posouvat, případně příslušnými gesty přiblížit a oddálit.

5.2.2 Rozložení pro počítač



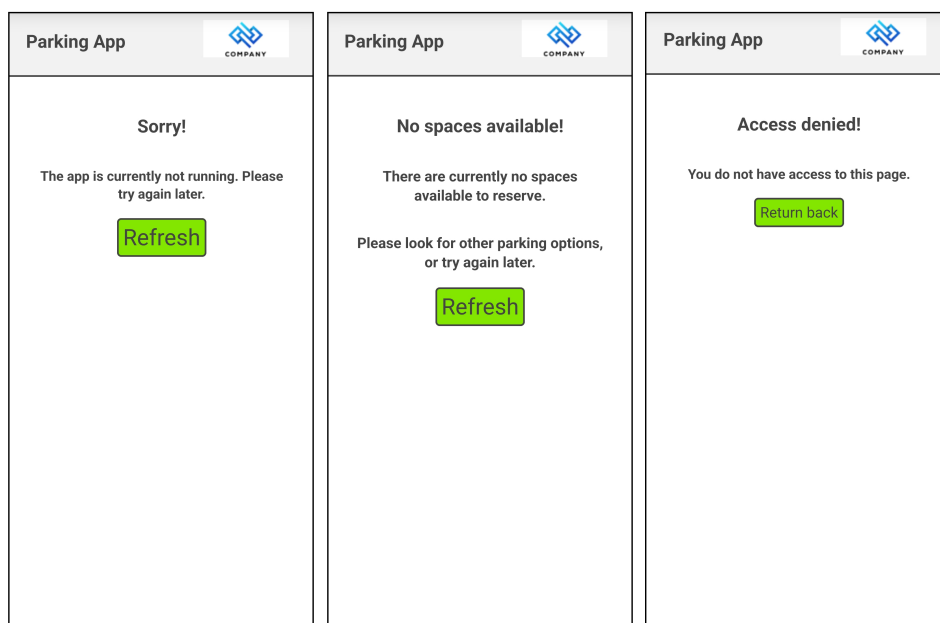
Obrázek 5.2: Rozložení základních obrazovek na počítači

Aplikace má funkční rozložení i pro počítač, viz. obrázek 5.2. To v podstatě vychází z rozložení pro mobilní telefon, ale snaží se více pracovat s prostorem do stran. Plánek parkoviště je přesunut doprava a objevila se tlačítka pro přiblížení a oddálení, která nahrazují gesta na mobilním telefonu.

5.2.3 Ostatní obrazovky

V případě nestandardní situace se mohou objevit varování, informující uživatele o stavu aplikace. Mezi ně patří:

- Neaktivní aplikace - Zobrazí se v případě, že aplikace daný den ještě nebyla správcem systému spuštěna. Uživatel je vyzván k trpělivosti. Na obrázku 5.3 vlevo.
- Plná kapacita - V případě, že se uživatel pokouší vytvořit rezervaci, nebo přichází na stránku ve chvíli, kdy jsou všechna dostupná místa zarezervována, je o této skutečnosti informován. Na obrázku 5.3 uprostřed.
- Přístup zamítnut - Pokud by se uživatel náhodou nebo záměrně dostal na odkaz vedoucí do správcovské konzole, proběhne kontrola, zdali se jeho



Obrázek 5.3: Obrazovky s různými druhy upozornění

emailová adresa nachází v seznamu správců. Pokud ne, je přesměrován na stránku s informací o neplatném přístupu. Na obrázku 5.3 vpravo.

5.2.4 Správcovské zobrazení

Hlavním prvkem správcovské konzole je tabulka na obrázku 5.4, která zobrazuje informace o všech manažerských místech. Barevné zvýraznění označuje stav parkovacího stání během zapnuté aplikace. Modře podbarvená jsou ta místa, jejichž majitelé mají na daný den schválenou absenci a tato stání se tedy nabízí k rezervaci pro ostatní zaměstnance. Když je rezervace vytvořena, místo zezelená a objeví se emailová adresa zaměstnance, který rezervaci vytvořil.

Každé místo je možné modifikovat. Kliknutím na tlačítko “manage,” nebo dvojitým poklepáním na celý řádek, se zobrazí dialogové okno (viz. obrázek 5.5), které správci dává dvě možnosti:

- “Exclude space” - Zapnutí tohoto příznaku značí, že místo nebude nikdy nabízeno, ani během majitelovy nepřítomnosti. Nastavení zůstane vypnuté nebo zapnuté až do doby, kdy se manuálně přepne zpět. Text takto označeného místa se v tabulce zbarví šedě.
- “Available today” - Podobná funkcionalita jako “Exclude space,” ale pouze na jeden den. Zvolením možnosti “Yes” pro “Ano” nebo “No” pro “Ne” lze určit, jestli se místo daný den bude zobrazovat v nabídce k rezervaci. Spolu s vypnutím systému ale nastavení zmizí a příště se bude opět řídit daty z docházkového systému.

Nr.	Owner's Name	Excluded	Available today	Reserved by	Manage
1	Otakar Tyrol	No	No		Manage
2	Olga Procházková	No	No		Manage
3	Jarmila Satahřková	No	Yes		Manage
4	Jilka Hogná	No	Yes		Manage
5	Jaroslava Malá	Yes	No		Manage
6	Oldřich Coufal	No	Yes		Manage
7	Tomáš Vitek	No	Yes		Manage
8	Miroslav Šudoma	No	Yes		Manage
9	Lukáš Krutý	No	Yes		Manage
10	Eva Zúchová	No	Yes	durisp4@fd.cvut.cz	Manage
11	František Šíp	No	Yes		Manage
12	Zuzana Bílá	No	Yes		Manage
13	Václav Růžicka	No	No		Manage
14	Jana Josišová	No	Yes		Manage
15	Jan Vondráš	No	Yes		Manage
16	Julius Novák	No	Yes		Manage
17	Adéla Čížová	No	Yes		Manage

tabulka pokračuje

Obrázek 5.4: Správcovské zobrazení umožňující úpravu údajů o parkovacích místech

Otakar Tyrol, Nr. 1

Exclude space No

Available today No

Cancel Save

Obrázek 5.5: Dialogové okno s možnostmi editace parkovacího stání

Tyto informace jsou zároveň pro připomenutí uživateli zobrazeny po najetí myši na symbol otazníku vlevo nad tabulkou, viz. obrázek 5.6.

Konzole dále umožňuje zapnout a vypnout celou aplikaci, nicméně uživateli je silně doporučeno nevypínat běžící aplikaci před koncem dne. To proto, že vypnutí systému zruší všechny vytvořené rezervace, na což nejsou zaměstnanci nijak upozorněni. Aplikace se sama vypíná automaticky na konci dne. Viz. obrázek

5.3 Shrnutí

Přístup do systému se dělí na dvě úrovně - obyčejnou a správcovskou. Běžný uživatel aplikaci využívá k vytvoření rezervace parkovacího místa, často z mobilního telefonu. Rozložení je tedy zaměřeno na pohodlné použití v mobilních zařízeních. Při práci je uživateli poskytována zpětná vazba o provedených

? **Color coding**

Nr. **Blue** - Blue background color means the parking space is available for reservation today, but is not yet reserved.

1 **Green** - Green background color means a reservation was already made for this parking space. The occupant's email is also displayed in the table.

2

3 **Gray text** - When the text is gray, it means that the parking space is set to be excluded.

4 **Table columns**

5 **Excluded** - This column indicates if the parking space should automatically be offered for reservations in the owner's absence. This setting is persistent and will stay this way until manually changed back. When set to **Yes**,

6 the parking space will NOT be offered even during the owner's absence, as indicated in the cominfo.

7

8

9 **Available today** - This column indicates if the parking space is being offered today (the owner is absent). Can be manually toggled between **Yes** and **No**, but this setting will be reset when the Parking App is turned off (or at the end of the day).

10

11

12

Obrázek 5.6: Nápořěda vysvětlující barevné návrhy správčovské tabulky

Warning!

This will irreversibly delete all reservations. Employees who made the reservations will NOT be notified about this.

The app does not have to be turned off manually, as it will happen automatically at the end of the day.

Are you sure you want to do this?

Obrázek 5.7: Upozornění varující uživatele před předčasným vypnutím aplikace

akcích v podobě informativních hlášek. Správci mají přístup jak k rezervaci míst, tak do správcovské konzole, která obsahuje seznam všech parkovacích a základní možnosti jejich spravování. Konzole je zaměřena na zobrazení na počítači, z důvodu rozměrů a nepřehlednosti na mobilním zařízení.

Kapitola 6

Testování

Stejně jako u tradičního softwaru, ověřovací a validační procesy mají za úkol kontrolu kvality implementace a odhalení deviací od požadavků uživatele. Dělí se na dva základní přístupy. Statické techniky se liší od dynamických tak, že nevyžadují “provedení” kódu (execution). Místo toho se zaměřují na artefakty, jako jsou návrhové dokumenty, například CSS, a zdroj HTML. Dynamické metody validace se zaměřují na prověřování systému zadáním vstupních dat a po proběhnutí porovnáním očekávaných výstupů se skutečnými[12].

Mezi konkrétní možnosti testování webových aplikací patří: [13]

- Funkční testování (Functional Testing)
- Test použitelnosti (Usability Testing)
- Testování výkonu (Performance Testing)

Funkční testování se zaměřuje na ověření toho, že systém skutečně umožňuje všechny funkcionality, které vyžaduje uživatel a že tyto funkcionality fungují správně[14]. Testování použitelnosti se týká hodnocení produktu nebo služby jejich testováním s reprezentativními uživateli. Obvykle se během testu účastníci pokusí dokončit typické úkoly, zatímco pozorovatelé sledují, poslouchají a dělají si poznámky. Cílem je identifikovat případné problémy s použitelností, shromáždit kvalitativní a kvantitativní data a zjistit spokojenost účastníka s produktem[15]. Testování výkonu je proces zhodnocení toho, jak systém funguje z hlediska odezvy a stability při určité pracovní zátěži. Testy výkonu se obvykle provádějí za účelem prověření rychlosti, robustnosti a spolehlivosti aplikace[16].

V rámci tohoto projektu proběhlo několik testovacích strategií. Konkrétně byly zvoleny některé z důvodu vnitřních předpisů firmy, jiné proto, že dávají z hlediska webového vývoje největší smysl. Jednotlivé strategie jsou podrobně rozepsány v sekcích 6.1 až 6.5.

6.1 Jednotkové testy

Jednotkové testy, anglicky “Unit Tests,” zahrnují testování softwarového kódu v jeho nejmenším funkčním bodě, tedy obvykle jedné třídě. Individuálním

testováním každé jednotky lze většinu chyb, které mohou být zaneseny do kódu v průběhu projektu, velmi brzy odhalit a zabránit jejich šíření do dalších částí projektu[17]. Jednotkové testy spouští funkce, které testují, s konkrétními vstupními parametry a kontrolují, že funkce vrátí očekávaný výstup. Například funkce sčítající dvě čísla by byla otestována tak, že by jí pomocí testu byla zadána dvě čísla, třeba 1 a 2 a očekávaný výsledek, tedy číslo 3. Pro každou funkci je třeba vytvořit takových testů více, aby se zkontrolovaly i krajní případy. Protože hlídat, že sčítací funkce vrátí správný výsledek při zadání dvou kladných nízkých čísel není příliš zajímavé. Problémy však už mohou nastat při snaze sečíst v naší funkci dvě nuly. Nebo dvě záporná čísla. Nebo záporné číslo a řetězec “abcd.” Je proto třeba ověřit nejen správné výsledky, ale i správné chování funkce v nestandardních situacích. To všechno jednotkové testy umožňují.

Pomocí těchto testů byly otestovány serverové funkce (ze souborů s příponou .gs, viz. sekce 4.3), které splňují dvě podmínky. Na vstupu přebírají jednu nebo více proměnných a vrací nějaký výstup. To jsou základní předpoklady pro použití jednotkových testů. Pro tento druh testování byla využita interní knihovna zadavatelské společnosti, která obsahuje všechny potřebné funkce ke spouštění vlastních konkrétních testů. Knihovna je soukromá v rámci firmy, proto ji nelze využít ve vzorové a anonymizované veřejné aplikaci. Více k tomuto tématu viz. kapitola 8

6.2 Tabulkové testy

Další knihovnou dostupnou v prostředí společnosti je knihovna tabulkových testů, Table Tests. Pomocí nich je možné ověřit, že zdrojové tabulky obsahují správná data. To přijde vhod v situacích, kde se data nahrávají z tabulek, které jsou zároveň jinými zaměstnanci firmy běžně k práci využívány. V takovém případě se snadno může stát, že správce tabulky náhodou nebo záměrně upraví buňky, které jsou aplikací využívány. To potom může způsobit problémy, zejména pokud se například číselná hodnota změní v jinou. Aplikace potom může zcela přestat fungovat. Je proto nutné zavést tyto kontroly zásadních hodnot. Knihovna zároveň nabízí možnost notifikace, pokud zachytí problém a tím umožňuje velmi rychlou nápravu chyby.

Zde je pomocí této knihovny kontrolována tabulka se seznamem zaměstnanců, jejich přidělených parkovacích míst a dalšími informacemi o parkovištích. Ta má však pro účely zavedení takovýchto kontrol omezení. Zejména v tom, že obsahuje i data navíc, která v aplikaci nejsou využívána a která mají jinou podobu. Nicméně však, protože jsou uložena ve stejné tabulce jako potřebná data, nedají se z důvodu způsobu implementace testovací knihovny z testů vyjmout. Tím jsou možnosti testování poměrně omezené, tyto testy proto byly využity alespoň pro taková data, která bylo možné zkontrolovat.

6.3 Výkonnostní testy

Přesto, že výkonnostní (performance) testy mohou být v řadě případů zanedbávány, jsou stále důležitou součástí vývoje a testování systémů[18]. Zde hraje výkon aplikace velkou roli, protože se předem očekává hlavní nápor uživatelů v danou chvíli, tedy ráno v pracovní dny. Aplikace byla proto otestována pomocí další interní firemní knihovny. Ta simulovala 200 přístupujících uživatelů naráz a jejich simultánní snahu o vytvoření rezervace na parkovací místo.

Z testu vyplynulo, že aplikace takový nápor vydržela bez problémů ohledně stability a rychlosti. Žádný z procesů, kterému se podařilo úspěšně vytvořit rezervaci, nezaznamenal problémy s dobou načítání a aplikace nespadla. Nicméně byl pozorován fenomén, že jednotlivé emailové adresy, se kterými se procesy připojovaly a které byly číselně seřazeny od 1 do 200, nebyly v seznamu rezervací uloženy v řadě. To je pravděpodobně zapříčiněno tím, že testovací knihovna neposílá jednotlivé procesy v řadě, tudíž se dopředu nedá zaručit, v jakém pořadí proběhnou. Každopádně i v případě, že by problém byl na straně aplikace, nebo Googlu, který serverové požadavky zpracovává, důležitějším výsledkem testu je, že se rezervace zapsaly a že aplikace nečekaně neskončila.

6.4 Revize kódu

Po dokončení implementace aplikace proběhla v rámci firmy kontrola kódu seniorním zaměstnancem, neboli Code Review. Bylo zkontrolováno především dodržování interních požadavků na kód, jako jsou názvy proměnných, souborová struktura nebo dostatečná dokumentace a komentáře. Dále revizor poskytl návrhy a tipy na optimalizaci aplikace, které byly do systému vloženy.

6.5 Testování přijetí uživatelem

Finální verze byla představena zástupcům z oddělení, pro které je aplikace určena a které zadávalo požadavky na její podobu, aby proběhlo testování přijetí, neboli User Acceptance Testing. Aplikace byla předána k vyzkoušení, nicméně v době odevzdání této práce ještě není známa zpětná vazba. Ta je důležitá pro zakomponování pozorování od samotných uživatelů.

6.6 Vyhodnocení testování

Testování ověřilo, že aplikace je stabilní a vyhodnocuje data správně. Jednotkové testování odhalilo chyby v některých funkcích, které díky tomu mohly být opraveny. Velmi důležité bylo ověřit, že aplikace vydrží očekávaný nápor až nižších stovek uživatelů. Zde nebyly odhaleny žádné zásadní problémy, byť není zaručeno udržení pořadí přístupujících zaměstnanců. Podstatné ale je,

že žádné rezervace nebyly po vytvoření přepsány jiným uživatelem a že se všechna dostupná místa naplnila. Dále je vhodné zmínit delší načítací doby při práci s aplikací. Důvodem je, že v každém kroku procesu vytvoření rezervace nebo změny nastavení parkovacího místa je nutné aktualizovat data z tabulek. To proto, aby se zamezilo konfliktům v případě souběžného přístupu více uživatelů ke stejné činnosti. Je proto nutné v každém kroku tabulku otevřít, načíst data a zkontrolovat, že nevznikne konflikt. To samozřejmě trvá. Některé načítací rychlosti by se daly zvýšit ukládáním určitých dat například do cache prohlížeče. Nicméně informací vhodných k takovému úkonu není mnoho, tedy by časová úspora nebyla tak veliká.

Kapitola 7

Závěr a shrnutí výsledků

Smyslem celého systému je zefektivnění využití parkovacích míst před budovou zadavatelské společnosti. Parkoviště disponuje dvěma druhy míst, přičemž první část jsou místa nepřirazená, na kterých může parkovat kdokoliv, a druhá část jsou místa dedikovaná, která jsou přirazená konkrétnímu zaměstnanci, nebo využívána za konkrétním účelem, například jsou na nich zaparkovány pracovní vozy.

Aplikace si klade za cíl umožnit zaměstnancům parkovat na těchto místech, na která běžně nemají přístup, v případě, že zrovna místa nejsou využívána. To aplikace zjišťuje ze seznamů spravovaných personálním oddělením - soupisem samotných míst a docházkovým listem. Jejich průnikem vzniká množina volných stání, která se mohou na daný den nabídnout k využití zaměstnancům přijíždějícím vlastním vozem. Součástí systému je návrh aplikace dle specifikací zadavatelského oddělení firmy.

Hlavním přínosem celého projektu je usnadnění procesu příjezdu do zaměstnání pro řadové zaměstnance, kteří ve firmě dlouhodobě trpí nedostatkem parkovacích míst. Možnost využít dedikovaná stání v případě jejich neobsazenosti je přínosná, zároveň vznik aplikace spravující parkovací místa otevírá další možnosti rozvoje a digitalizace v tomto směru, které by přinesly další výhody.

Systém byl implementován podle vytvořené specifikace a následně byl podroben několika různým druhům testování, které ověřily jeho robustnost a spolehlivost. Díky této kontrole je možné systém bezpečně nasadit do běžného provozu. Aplikace splňuje základní funkční požadavek, který upravuje zajištění systému parkovacích míst a umožňuje zaměstnancům využít volná stání, která pro ně doteď byla mimo dosah. Přidaná hodnota spočívá v zahájení možného rozvoje digitalizace celého parkoviště. Aktuálně aplikace pracuje výlučně s téměř 50 parkovacími místy, která jsou dedikována konkrétnímu uživateli. Celé parkoviště ale obsahuje přes 200 stání. Je tedy nyní možné zaměřit se například i na tato místa a využít dodatečné systémy, například kamerový, k zjištění stavu obsazenosti a jeho následné prezentaci zaměstnancům. Systém proto vytváří prostor k aktualizaci běžného přístupu k parkování a k jeho rozvoji v digitálním světě.

7.1 Možnosti rozvoje

Aplikace má několik funkcí, které by v zájmu zvýšení uživatelského pohodlí bylo možné rozšířit. V první verzi byla upřednostněna jednodušší verze aplikace, která uživatelům umožní snazší seznámení s jejím používáním. Přidání funkcionalit firma plánuje na pozdější dobu, kdy už bude aplikace zavedena v běžném režimu.

Mezi možná rozšíření uživatelského prostředí patří možnost filtrovat výsledky ve správcovské tabulce podle jednotlivých sloupců - jméno, dostupnost, stav rezervovanosti. Také je možné rozšířit funkcionalitu o dlouhodobější přehled nebo export statistik obsazenosti a rezervovanosti všech míst.

Běžný uživatel by mohl mít zájem o zaslání informací o stavu jeho rezervace emailem. S tím by se dalo spojit i zaslání zprávy informující o zrušení rezervace správcem systému.

Tato i všechna další rozšíření nicméně přichází v úvahu až po zakomponování aplikace do běžného života zaměstnanců firmy. Všechny takové změny by také musely být konzultovány a schváleny se zákazníkem, tedy zadavatelským oddělením firmy.



Kapitola 8

Příloha

Z důvodu obtížnějšího přístupu k samotné aplikaci, která byla vytvořena a je využívána v rámci uzavřeného systému zadavatelské firmy, byla vytvořena její kopie s anonymizovanými daty. Ta je veřejně dostupná [19]. Z důvodu nezrušitelného automatického přesměrování ze strany společnosti Google je nemožné se do aplikace dostat, pokud je uživatel v prohlížeči přihlášen více než jedním účtem. Nejjednodušší způsob, jak se v takovém případě do aplikace vstoupit, je přes "Anonymní režim." Tím se vyloučí nutnost odhlášení ze všech účtů Google v prohlížeči, což je samozřejmě další možnost. Přístup do správcovské konzole je i v této aplikaci vázán na konkrétní emailovou adresu, proto je pro běžného uživatele skrytý. Přístup k originálu je možný po osobní dohodě s kontaktní osobou zadavatele: Ing. Jakub Hudec, jakub.hudec@valeo.com.



Bibliografie

- [1] Carlton Reid. *Roads Were Not Built for Cars: How Cyclists were the First to Push for Good Roads & Became the Pioneers of Motoring*. Washington, DC: Island Press/Center for Resource Economics, 2015. ISBN: 978-1-61091-688-2. DOI: 10.5822/978-1-61091-688-2_6.
- [2] Jakub Ďuriš. „Optimalizace využití parkovacích míst. Semestrální projekt“. Led. 2023.
- [3] Michael Maguire. *Google Sheets programming with Google Apps Script*. 2015.
- [4] *DATA INTEGRATION GLOSSARY*. Tech. zpr. 400 Seventh Street, S.W., HIAM-30, Washington, DC 20590: US Department of Transportation, srp. 2001, s. 10.
- [5] W3Schools. *C Multidimensional Arrays*. URL: https://www.w3schools.com/c/c_arrays_multi.php.
- [6] Mike Watson. *Managing Smaller Projects: A Practical Approach (Project Management Essentials Library)*. 2. Multi-Media Publications Inc., 2006, s. 117. ISBN: 9781895186857.
- [7] Lawrence Chung et al. „Non-Functional Requirements in Software Engineering“. In: Boston, MA: Springer US, 2000. ISBN: 978-1-4615-5269-7. DOI: 10.1007/978-1-4615-5269-7_1. URL: https://doi.org/10.1007/978-1-4615-5269-7_1.
- [8] James Ferreira. *Google Apps Script Web Application Development Essentials*. 2. O'Reilly Media, 2014, s. 3. ISBN: 9781491946183.
- [9] Google. *Google Apps Script overview*. URL: <https://developers.google.com/apps-script/overview>.
- [10] Margaret Reid-Miller. *Two-Dimensional Arrays*. Pittsburgh, PA, 2010. URL: <https://www.cs.cmu.edu/~mrmiller/15-110/Handouts/arrays2D.pdf>.
- [11] Google. *Web Apps*. URL: <https://developers.google.com/apps-script/guides/web>.

- [12] F. Ricca a P. Tonella. „Analysis and testing of Web applications“. In: *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001*. 2001. DOI: 10.1109/ICSE.2001.919078.
- [13] Shakti Kundu. *Web Testing: Tool, Challenges and Methods*. Břez. 2012. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1%5C&type=pdf%5C&doi=315b1af595f5e622013a02a3d46196d1d602b8f4>.
- [14] S. Nidhra. „Black Box and White Box Testing Techniques - A Literature Review“. In: *International Journal of Embedded Systems and Applications 2* (čvn. 2012). DOI: 10.5121/ijesa.2012.2204.
- [15] Usability.gov. *Usability Testing*. URL: <https://www.usability.gov/how-to-and-tools/methods/usability-testing.html>.
- [16] Tricentis. *Performance testing, best practices, metrics & more*. URL: <https://www.tricentis.com/learn/performance-testing>.
- [17] Dorota Huizinga a Adam Kolawa. *Automated Defect Prevention Best Practices in Software Management*. 1. Wiley-IEEE Computer Society Pr, 2007. ISBN: 9780470042120.
- [18] Filippos I. Vokolos a Elaine J. Weyuker. *Performance Testing of Software Systems*. 1998. URL: <https://dl.acm.org/doi/pdf/10.1145/287318.287337>.
- [19] Jakub Ďuriš. *Parking App*. URL: https://script.google.com/macros/s/AKfycbwylbUUEGAV3RG0CLmZ-3qsaIyu8_Z0Tk0s_2vQQ9VDbU2AbQm0JtNFcx8G1_mjmm-V6g/exec.