

Smart Time Management App

Technická zpráva k projektu do předmětu ITU
FIT VUT v Brně, 2020

Název týmu

Tým s podporou hluboké umělé umělé inteligence s nízkým IQ(+0)

Autoři

Filip Osvald, xosval04

Daniel Karásek, xkaras38

David Hudák, xhudak03

Obsah

1. Zadání a organizace týmu

1.1 Cíl

1.2 Tým

1.3 Roadmapa

1.4 Rizika a opatření

2. Průzkum a zkušenosti

2.1 Existující řešení

Microsoft To Do (Filip, Daniel, David)

Google kalendář, kalendář Huawei (Daniel, Filip)

Google kalendář, kalendář Samsung (David, Daniel)

Moje Záměry (David)

2.2 Uživatelské potřeby

2.3 Shrnutí

3. Architektura řešení

3.1 Architektura systému

3.2 Architektura aplikace/í

3.3 Datový model

3.4 Vybrané technologie a implementaci

4. Návrh GUI - aplikace pro pc [xosval04]

4.1 Požadavky na GUI

4.2 Makety

4.3 Pilotní test

4.4 Vyhodnocení testu a revize návrhu

5. Návrh GUI - Klikací část Android aplikace [xhudak03]

5.1 Požadavky na GUI

5.2 Makety

5.3 Pilotní test

5.4 Vyhodnocení testu a revize návrhu

6. Návrh GUI - zobrazení kalendáře a úprava událostí, android - [xkaras38]

6.1 Požadavky na GUI

6.2 Makety

6.3 Pilotní test

6.4 Vyhodnocení testu a revize návrhu

7. Implementace GUI - mobilní aplikace Chleba [xhudak03]

7.1 Implementace

7.2 Použité nástroje a knihovny

7.3 Finální testování

7.4 Vyhodnocení testu

[8. Implementace GUI - nastavení mobilní aplikace \[xkaras38\]](#)

[8.1 Implementace](#)

[8.2 Použité nástroje a knihovny](#)

[8.3 Finální testování](#)

[8.4 Vyhodnocení testu](#)

[9. Implementace GUI - desktopová aplikace \[xosval04\]](#)

[9.1 Implementace](#)

[9.2 Použité nástroje a knihovny](#)

[9.3 Finální testování](#)

[9.4 Vyhodnocení testu](#)

[10. Závěr \[xhudak03\]](#)

[11. Závěr \[xosval04\]](#)

[12. Závěr \[xkaras38\]](#)

[Reference](#)

1. Zadání a organizace týmu

1.1 Cíl

Cílem projektu je vytvořit aplikaci na efektivní správu času. Uživatel bude moci spravovat jednotlivé události. Nastavit si, kdy má být na událost upozornění, přidat k ní informace, určit její typ etc. Aplikace bude navíc obsahovat navíc sadu denních motivačních citátů. Má pomoci obecně lidem, kteří jsou zdušeni uspěchaností a chaosem moderní doby. Má jim pomoci přehledně organizovat svůj čas bez vydání velkého psychického úsilí. Toho je docíleno možností zadávat data inovativním způsobem - používáním šablon událostí (primárně v mobilní verzi). Na desktopové aplikaci uživateli pak pomáhá chytré řazení událostí.

1.2 Tým

V týmu jsou 3 (David, Daniel, Filip) kolegové, přátelé a podporovatelé moderních technologií. Všichni budou dělat cca všechno, jelikož budeme pracovat metodami extrémního programování.

Výsledné rozdělení práce pak bylo následující:

- David - Vedoucí mobilní aplikace
- Filip - Vedoucí desktopové aplikace
- Daniel - Hlavní tvůrce modelu, vývojář features pro obě platformy

1.3 Roadmapa

Kompletní návrh (všichni) - 27. 10. 2020.

Hotový backend pro obě platformy (všichni) - 2. 11. 2020

Připravená aplikace pro platformu android (všichni) 19. 11. 2020

Hotová aplikace včetně testování pro platformu android (všichni) - 20. 11. 2020

Připravená aplikace pro platformu desktop (všichni) 28. 11. 2020

Hotová aplikace včetně testování pro platformu desktop (všichni) - 29. 11. 2020

Kompletní zpráva (všichni) - 1. 12. 2020

1.4 Rizika a opatření

V těchto náročných časech se vyskytují větší zdravotní rizika pro členy v týmu. Při úmrtí člena týmu by nastal vážný problém se stíháním termínů a přerozdělením práce. Proto každý člen týmu povinně udržuje správnou hladinu vitamínů v těle a pravidelně si měří teplotu. Člen týmu by si také mohl zvolit příliš náročný cíl který by jej, případně i ostatní členy

týmu příliš zdržel v práci na zbytku řešení. Dále je třeba udržet efektivní komunikace v týmu. Proto jsme zvolili jako komunikační platformu Discord který je pro všechny členy týmu nejvhodnější.

2. Průzkum a zkušenosti

2.1 Existující řešení

V následujících podkapitolách padnou informace ohledně našeho průzkumu trhu s podobnými aplikacemi.

Microsoft To Do (Filip, Daniel, David)

Aplikace pomůže ujasnit si cíle a dá vám dobrý pocit ze splnění cíle ale nepomůže s organizací času. Neumožňuje filtrování dle kategorií a priorit a má nevyhovující grafické rozhraní a bez motivačních citátů nemá uživatel vždy dostatečnou motivaci ke splnění úkolů.

Google kalendář, kalendář Huawei (Daniel, Filip)

Provedl jsem analýzu aplikací, které jsem dříve používal na svém telefonu. Ani jedna nepodporovala to, co bych od tohoto typu aplikace očekával. Přesto jsou zde záchránky inspirativních features. Jejich pozitivum bylo snadné zadávání a přehledné zobrazení dat. Negativem byla nedostatečná podpora pro pomoc z dodržováním zdravého životního stylu a nízká diverzita v možnostech zadávání dat.

Google kalendář, kalendář Samsung (David, Daniel)

Důsledně jsem prozkoumal aplikace, které běžně používám na svém telefonu. Ani jedna nepodporuje dostatečně to, co jsme si slíbili v zadání, tudíž je řešení poněkud slabší. Na druhou stranu je v některých ohledech inspirativní. V Samsung kalendáři se mi líbí jeho poněkud konzervativní vzhled a intuitivní jednoduché zadávání. V Google kalendáři naopak oceňuji jeho kreativní zpracování a zvýraznění jenom těch dat (od slova datum), která jsou podstatná nějakou poznámkou. Problémem je, že ani jedna z těchto aplikací nenabízí úplně ideální organizátor času, jelikož zde není úplně optimální zpracování jednotlivých dnů.

Oproti oběma aplikacím bych tedy doporučil zapracovat na organizaci jednotlivého dne, ideálně s nějakou snadno posunovací intuitivní nabídkou a tvorbou bloků jednotlivých činností. Dobré by bylo přidat blok nějaké generické činnosti (zábava, práce, studium, jídlo atp.) Co se mi líbí je, že s kamarádem sdílím některé události, které se týkají školy, tedy nemusím jenom já zadávat všechny činnosti a v případě snadné implementace bych takovou funkcionalitu podpořil.

Moje Záležitostí (David)

Jako poslední aplikace na průzkum byla zvolena aplikace Moje Záležitostí (Google zřejmě neprovedl správný překlad). Ta je poměrně intuitivní, má vícere možnosti ručního zadávání, přehledné zobrazení dní, ale i jednoduchý kalendář k tomu. Problémem je nižší výkon aplikace a zobrazované reklamy. Osobně si nemyslím, že je pro správnou efektivitu práce vhodné mít v aplikaci jakékoliv reklamy a náš SW by měl být bez nich buď zcela zdarma (freeware), nebo zpoplatněn nějakou jinou cestou. Věc, která by pak podle mě měla chybět úplně, je paywall na lepší funkce - aplikace by měla být od začátku zcela funkční.

2.2 Uživatelské potřeby

I přes omezené možnosti komunikace (viz koronavirová pandemie) jsme zjistili několik postřehů k potřebám uživatelů a k možnostem, které by mohla poskytovat naše aplikace.

Největším problémem dneška pro spoustu studentů a zaměstnanců je home office. Ten je na jednu stranu zajímavý v tom, že člověk pracuje z domácího prostředí, ale je zároveň velmi nebezpečný pro naši produktivitu. Sami na sobě vidíme, že špatně organizujeme čas. Například jsme sice ušetřili času na cestě do školy, ale o to více času utrácíme v neefektivní práci s naším časem. Například čas, který bychom strávili běžně u přednášek ve škole, teď můžeme věnovat tomu, že přednášku zastavíme, případně odložíme její zhlédnutí na později a věnujeme se v dané chvíli jiným věcem. Problémem pak je, že už si pak správně nenaplánujeme čas, kdy se skutečně na danou přednášku podíváme a zjistíme, že tak nějak nezbyvá čas kvůli jiným úkolům a povinnostem. To samé platí i pro lidi v home office (například naši rodiče) - ti sice mají mnohem volnější ruku v tom, jak si organizovat své činnosti, na druhou stranu pak také občas nestihnou udělat vše, co měli, v nějaké přijatelné době. Podobné problémy s organizací se pak týkají i lidí v managementu, ale obzvlášť v této situaci, i lidí ve zdravotnictví. Viz příklad nutnosti scházet se v 10 večer v uzavřené hospodě.

Pro zjištění takových uživatelských potřeb jsme našli někoho, kdo s vývojem SW nemá nic společného a sám by takovou aplikaci uvítal. Daná osoba zůstane anonymní.

Dotazovaný projevil zájem nad danou aplikací, kde vznesl pro nás nečekané požadavky. Velmi by uvítal zdůrazňování plateb k uhrazení, nějaký jejich přehledný souhrn a další věci, které s tím souvisejí, protože ty jsou pro dobrého manažera skoro až esenciální. Dále zdůraznil nutnost možnosti zapsat si schůzky a v případě jejich kolizí alespoň nějaké upozornění. Obecně pak velmi směřoval k tomu, že daná aplikace by měla pomáhat s upozorňováním na důležité činnosti a akce.

2.3 Shrnutí

Dle našich závěrů vyplývá, že aplikace by měla být jednak jednoduchá na používání, měla by být co nejvíce uživatelsky přívětivá a měla by mít velké možnosti, co se týká zadávání informací. Také by měla být stabilní, bez reklam, bez sekání i pro méně výkonná zařízení. Měla by také upozorňovat na prioritní činnosti, jako jsou platby atp.

Z nalezených informací také jasně vyplývá, že je naše aplikace potřeba a může zachránit čas, životy, ale i nešťastné ministry.

3. Architektura řešení

3.1 Architektura systému

Aplikace bude sestávat z backendu a frontendu, kde backend bude nějaká knihovna funkcí spravující data a upozornění a bude mít komunikační rozhraní pro frontendovou část reprezentující GUI. K aplikaci budou ještě pomocné soubory s důležitými daty, ze kterých bude backend číst. V případě rozšíření se bude používat nějaký databázový systém, který by zajišťoval sdílení dat mezi více zařízeními/aplikacemi.

//Jaké části bude mít celý systém? Jak budou komunikovat? Co bude která část dělat?

3.2 Architektura aplikace/í

Architektura aplikace je dle vzoru MVC. Modelem budou nějaké soubory s daty (zřejmě typově .txt, budou obsahovat záznamy událostí a vybrané motivační citáty), viewem bude GUI aplikace, tedy zobrazení dat. Bude obsahovat tlačítka, vstupní bloky apod., které vyvolají odpovídající callback funkce controlleru. Ten poté změní model, aby odpovídal požadovaným změnám uživatele.

V průběhu vývoje došlo k menší změně v přístupu k datům, jelikož Xamarin a WPF/WinForms používají místo controlleru spíše ViewModel. Proto byl model MVC nahrazen modelem ModelViewModelView. Model zde zajišťuje práci s daty aplikace (ukládání událostí do JSON souboru, jejich načítání a parsování atp.) ViewModel zajišťuje uchovávání mezidat při komunikaci Modelu a View. View zůstává totožný s původní myšlenkou, tedy obsahuje tlačítka, výpisové texty, kalendáře atp.

3.3 Datový model

Do základu budeme potřebovat s daty (od slova datum) propojené události, které se budou skládat z typu události, doby trvání, popisku, priority a případně nějaký dalších vedlejších dat. Také se budou ukládat záznamy o správné životosprávě, tedy časy jídla, pití atp. v rámci dne. Dále budou konstantní soubor s vybranými motivačními citáty, z něhož se budou vybírat ty nejvhodnější pro aktuální den. K tomu všemu bude pomocný soubor zahrnující statistiku činností. K přístupu k těmto datům budou sloužit odpovídající funkce API, například nahraj činnosti ze souboru, vyber motivační citát, přečti aktuální akutní upozornění.

V průběhu implementace byl zvolen pro datový model formát JSON.

3.4 Vybrané technologie a implementaci

Plánovanou technologií pro backend je C# a pro implementaci GUI na PC předpokládáme Windows Forms/Windows Presentation Forms v rámci VS a na Android Xamarin se C#. Pro

sdílení kódu GitHub a pro sdílení ostatních souvisejících informací bude využit Google Drive, případně nějaké ad hoc technologie.

4. Návrh GUI - aplikace pro pc [xosval04]

4.1 Požadavky na GUI

Základním a logickým požadavkem na GUI je, aby bylo snadné, intuitivní a pochopitelné pro cílovou skupinu. Tedy mělo by být snadno možné přidávat události do jednotlivých dní, rychle si je zobrazit, ale také by se s nimi mělo dát jednoduše interagovat, natahovat je, upravovat je, odstraňovat je, přesouvat je atp. Obecně by nemělo být časově náročné vytvořit nějakou událost, aby se z organizátoru času nestala spíš překážka.

Bude realizovat celodenní časové plány uživatelů, zobrazovat je, upozorňovat a sám se zasazovat o lepší organizaci času.

a) PC

b) Android

Zde se zaměříme na část pro pc.

4.2 Makety

Kalendář	Upozornění, doporučení		Nastavení
	Událost, jméno, typ, popis, priorita		další
			-
			+
0-24	Motivační citát		

4.3 Pilotní test

Test proběhl s wireframy na rodinném příslušníkovi jelikož kvůli nouzové situaci je počet možných uživatelů pro pilotní test omezený. Uživatel měl za úkol přidat, smazat a vyhledat události. Dále byl tázán které prvky mu chybí, které by změnil a které by odstranil.

4.4 Vyhodnocení testu a revize návrhu

Uživatel vše intuitivně zvládnul a zejména ocenil velikost tlačítek při jeho zrakovém omezení způsobené věkem. Jediná nejasnost byla přehled událostí dle hodin která je v wireframu zatím označená jako 0-24. Uživatele také zaujaly motivační citáty které bohužel není možné zobrazit ve wireframu. Poslední bylo zmíněné tlačítko minus které uživatel narhнул změnit na nápis smazat.

5. Návrh GUI - Klikací část Android aplikace [xhudak03]

Tato část se věnuje mnou vypracovanému návrhu GUI.

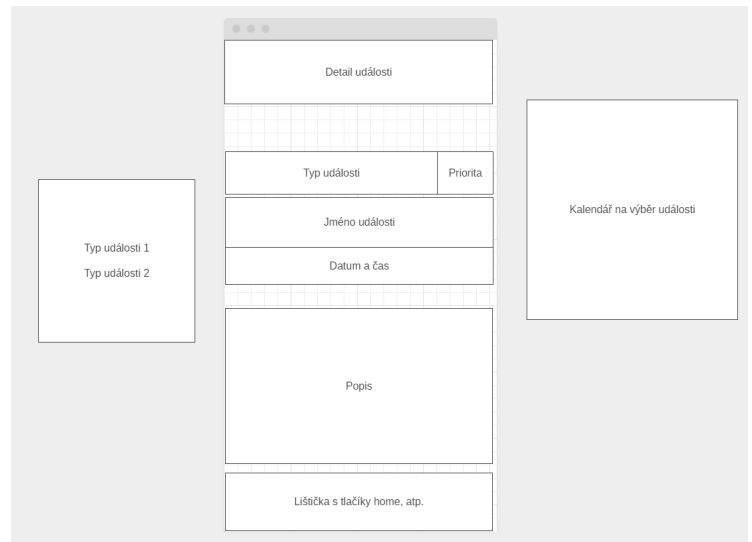
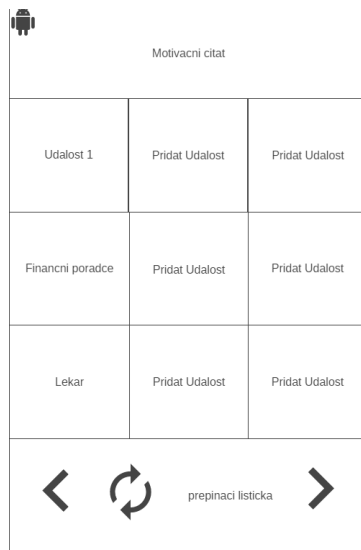
5.1 Požadavky na GUI

Základním a logickým požadavkem na GUI je, aby bylo snadné, intuitivní a pochopitelné pro cílovou skupinu. Tedy mělo by být snadno možné přidávat události do jednotlivých dní, rychle si je zobrazit, ale také by se s nimi mělo dát jednoduše interagovat, natahovat je, upravovat je, odstraňovat je, přesouvat je atp. Obecně by nemělo být časově náročné vytvořit nějakou událost, aby se z organizátoru času nestala spíš překážka.

Bude realizovat celodenní časové plány uživatelů, zobrazovat je, upozorňovat a sám se zasazovat o lepší organizaci času.

Konkrétně v mé části naší aplikace se věnuji klikací části (nikoliv kalendářní). Základním poznatkem z hlediska telefonů je, že důležité ovládací prvky by se měly nacházet spíš níže na displeji a nahoře by se měla zobrazovat data. Proto by se měly nahoře vypisovat blížící se události a vybrané motivační citáty a na nižších pozicích by měly být hlavní ovládací prvky. Ideou této části aplikace je, že přidání událostí (či provedených činností) by mělo být ideálně co nejrychlejší, a proto bude mít design inspirovaný Windows Phonem. Tedy člověk klikne jenom na některou z uživatelsky upravených dlaždic a rovnou přidá událost (pokud se bude jednat o dodržení pitného režimu či stravy), nebo otevře nabídku na přidání například upozornění na platbu. Rovnou budou některá políčka formuláře dle tohoto požadavku předvybraná a uživatel tak ušetří čas prozkoumáváním nabídky.

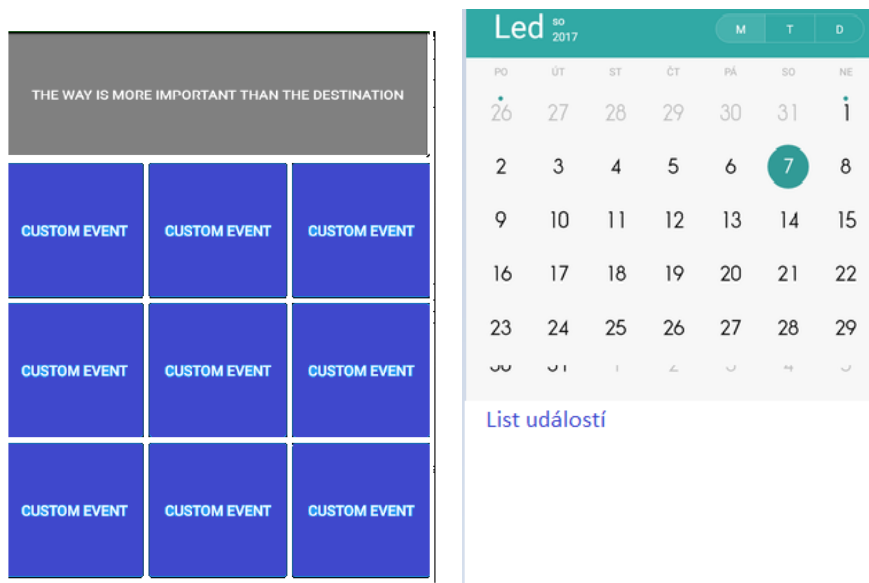
5.2 Makety



Mírná inspirace vzniká u již zaniklého prostředí operačního systému Windows Phone viz obrázků níže.



Dále jsou zde obrázky z prvních fází vývoje před implementací jakékoliv funkcionality (žádné přechody mezi okny, žádná funkcionální, pouze okna).



5.3 Pilotní test

Z důvodu celostátní karantény nebylo možno provést pilotní test na jakékoliv osobě v mém okolí, nýbrž se okruh zmenšil pouze na mou matku a jejího kolegu z práce (pracují v ekonomickém sektoru ve zdravotnictví, tudíž bylo logické je zvolit jako vhodné testery).

Jelikož se nejedná o wireframe celé aplikace, za úkol měli pouze přidat nějakou událost, rozkliknout aktuálně zobrazenou událost/motivační citát, přidat událost a u druhého wireframu správně vyplnit hodnoty u detailu události. Náš návrh aplikace byl zřejmě správný, tedy přidání události a následně jeho vyplnění proběhlo bez problému. Jediné drobné nepochopení bylo u určování typu událostí a jejich priorit, ale následně bylo uznáno, že to dává smysl z hlediska optimalizace času a následného backendu (neřekli přímo slovo backend). Pilotní test tedy proběhl úspěšně.

5.4 Vyhodnocení testu a revize návrhu

Dle reakcí testovacích subjektů (z důvodu epidemie proběhlo testování velmi omezeně) se dá usoudit, že návrh proběhl správně a s citem a že se následně bude moct dát pokročit k mockupu a nakonec i k vývoji výsledného GUI dle původního wireframu.

6. Návrh GUI - zobrazení kalendáře a úprava událostí, android - [xkaras38]

6.1 Požadavky na GUI

Základním a logickým požadavkem na GUI je, aby bylo snadné, intuitivní a pochopitelné pro cílovou skupinu. Tedy mělo by být snadno možné přidávat události do jednotlivých dnů, rychle si je zobrazit, ale také by se s nimi mělo dát jednoduše interagovat, natahovat je,

upravovat je, odstraňovat je, přesouvat je atp. Obecně by nemělo být časově náročné vytvořit nějakou událost, aby se z organizátoru času nestala spíš překážka.

Tato část gui se zaměřuje na zobrazení dat a jejich editaci v Androidu. V zobrazovacím gui člověk může vidět současný den a kalendář pro daný měsíc s podbarvenými dny, ve kterých se nachází událost. Bude přirozeně možné přecházet do dalších nebo minulých měsíců. Při kliknutí na konkrétní den se zobrazí detail dne. Ten bude obsahovat jednotlivé události po hodinách. Po rozkliknutí konkrétní události se ukáží její detaily. Ty bude následně možné i editovat.

6.2 Makety



6.3 Pilotní test

Pilotní test byl proveden na rodinných příslušnících. Ti měli za úkol provést několik základních úkonů jako je upravit nebo smazat událost. Testeři byli nejdříve tázáni na celkový dojem. Poté na to zda je gui intuitivní a přehledné. Nakonec jsem se ptal jestli jim nechybí nebo nepřebývá nějaká funkcionality.

6.4 Vyhodnocení testu a revize návrhu

Uživatelé všechny úkony bez problémů zvládli. Chybělo jim lépe umístěné tlačítko na smazání události. Na druhou stranu ocenili velikost tlačítek a nepřepĺácanost zbytečnými funkcemi.

7. Implementace GUI - mobilní aplikace Chleba [xhudak03]

7.1 Implementace

Základní částí aplikace je view obsahující všechna důležitá okna. Základním rozcestníkem je zde část AppShell, která obstarává přechody mezi hlavními okny. Ty jsou celkem tři - tzv. WindowsPhonePage (inspirována operačním systémem Windows Phone), CalendarPage, obsahující kalendář a výpis všech událostí daného měsíce s možností přidávat události na konkrétní dny, a Settings, který obsahuje prostor pro nastavení šablon a pro případné smazání všech dat. Kromě těchto hlavních oken aplikace ještě obsahuje okna pro správu šablon (přístupná přes settings), okno přidání události a okno detailu události (přístupné přes předchozí dvě hlavní stránky).

Celý view pracuje se svým code behindem, kde volá asynchronní funkce pro přesuny v oknech a s bindingy na viewModel. Pro vykonání operace z viewModelu se používá praktika Command. Samotný viewModel pak sestává především z DataFlowViewModel, který, jak název napovídá, zprostředkovává veškeré přesuny dat mezi modelem a viewem a zajišťuje potřebné dodání informací pro view model. ViewModel také pracuje s listViewem, který je obsažen v kalendářové stránce a vypisuje všechny události daného dne/daného měsíce (záleží, jestli uživatel vybírá den nebo měsíc). ListView je pak navázán na třídu ListViewModel, která obsahuje všechny komponenty potřebné pro správné fungování listView. Především obsahuje název události, den a čas a jeho prioritu - ovšem pouze ve formě barvy. Poslední důležitou komponentou, kterou obsahuje, je ID události, kterou vypisuje. To se používá pro případné upravování, prohlížení a odstraňování událostí.

Srdcem celé aplikace je pak model, který obsahuje funkce pro práci s JSON souborem, který obsahuje veškerá důležitá data pro fungování aplikace, tedy záznamy jednotlivých událostí a 9 šablon. Umožňuje výpis několika prvních událostí, výpis všech událostí měsíce, výpis událostí daného dne, odstranění události na základě ID či změnu události na základě ID a přijatých alternativních informací. Model pak mimo jiné ještě obsahuje třídu s názvem Event. Ta je v podstatě pouze strukturou pro záznam jednotlivých událostí a obsahuje informace jako je název události, priorita, čas, datum, popis atp.

7.2 Použité nástroje a knihovny

K implementaci GUI mobilní aplikace byla využita technologie Xamarin běžící v prostředí Visual Studio. Z mé strany bohužel musím konstatovat, že výběr nebyl v mnoha ohledech

úplně šťastný. Za prvé, Visual Studio je opravdu náročné na kompilaci a Android Studio, které používá na emulování aplikace je též velmi náročné a trvá to někdy i dlouhé minuty, než se načte. Další poměrně problematickou věcí Xamarinu je řešení updatování stránek - to se povedlo vyřešit až s pomocí přepsání metody stránky OnAppearing. Další negativní vlastností vybraných technologií bylo velice špatné zobrazování na menších displejích při vývoji (vývoj probíhal z větší části na notebooku, kde je monitor menší). Co vnímám jako výhodu zvoleného řešení, je poměrně interaktivní debugger, kde se dalo snadno v kolonce watches vybrat všechny proměnné, které jsem při debugingu zrovna chtěl sledovat.

Na knihovnu byla využita knihovna Xamarin Forms Controls Calendar stažená jako doplněk NuGet. Její design není příliš pokrokový, ale jako jednoduché API pro kalendář se poměrně osvědčila. Jediným výrazným problémem bylo zajišťování, že aplikace nespadne při nevybraném datu.

Co se týká obecných věcí okolo designu Xamarinu, tak tam jsou mé vývojářské pocity také poněkud rozporuplné. Na jednu stranu vytvořit celkem designově pěkné GUI nebylo náročné, na druhou stranu nebylo možné využívat nějaké interaktivní vývojářské nástroje a vše jsem tak v podstatě musel psát ručně, což mi oproti jiným známým technologiím příliš nevyhovovalo.

Zajímavou problematikou byla komponenta ListView. U ní byl dlouhodobě problém s nemožností ji aktualizovat, pokud došlo k přidání do seznamu v jiném okně aplikace než v místě, kde byl list vykreslován. To se povedlo vyřešit až s pomocí přepisované metody OnAppearing, která v podstatě volá vybrané funkce kdykoliv, když se aplikace přesune na dané okno.

7.3 Finální testování

Testování z koronavirových důvodů neprobíhalo úplně ve formě, jakou bychom si přáli, avšak nutno říci, že k žádaným informacím jsme se dostali. Testování probíhalo přes rozesílání apk souboru s aplikací, kdy tázaný dostal toto apk a použil jej u sebe na testování s tím, že měl od nás zadané instrukce. Testování v první řadě (bohužel) odhalilo některé implementační chyby v řešení, například pády aplikace při přidání události s neoznačeným datem. Proto pak muselo dojít k druhé fázi distribuce testovací aplikace, kde už tento nedostatek byl upraven.

Během druhé fáze testování byly odhaleny další drobné nedostatky (jako například zvedání priority při prohlížení události, neaktualizující se seznam událostí při změně data atp.) Ty byly poměrně rychle vyřešeny. Nedostatkem tohoto testování pak bylo, že se uživatel až příliš s námi zabýval opravou a hledání implementačních chyb než samotným uživatelským rozhraním, takže bylo nutno zvolit další osobu k testování.

Dalším uživatelem na testování byl můj kamarád z gymnázia, který se poměrně výrazně zajímá o telefony, takže byl velmi ochotný při pomoci s testováním, jak to jenom bylo možné. Bohužel tato osoba vlastní poměrně moderní zařízení s Android 11, což zatím plně nepodporují vývojářské prostředky Microsoftu, takže u uživatele nastal bug se špatným vykreslováním událostí. Proto bylo přikročeno k poněkud nekonvenční metodě, kdy jsem s uživatelem sdílel obrazovku a hovor a nechal jsem uživatele (mého kamaráda) mě navádět v

mém prostředí. Překvapivě se chytal poměrně rychle a zvládl mě s vlastní zvědavostí navést k vyřešení všech úkolů (přidat událost, zobrazit událost, přidat událost přes kalendář, najít událost v kalendáři). Uživatel vyjádřil přání možnost v kalendáři seřadit data podle priority případně vyfiltrovat nižší priority pryč. Líbila by se mu také možnost ukázat pouze události daného typu. Například pouze platby nebo nadcházející narozeniny.

7.4 Vyhodnocení testu

Testování s uživateli ukázalo, že uživatelé mají větší tendenci najít chyby než původní vývojáři, což bylo na jednu stranu přínosné, na druhou stranu v tom vznikala škoda toho, že se až příliš dostávali k implementačním detailům dané aplikace. U druhého uživatele pak nastal problém s nedostatečnou kompatibilitou s Android 11. Naštěstí se u druhého uživatele podařilo najít náhradní způsob testování, který proběhl úspěšně. Určitě bych se mnohem více zaměřil na možnost filtrování událostí v kalendáři, ať dle priorit typů nebo jiných predikátů. Dále bych určitě zvážil implementaci upozornění (například s možností zvolit si nějakou melodii při blížící se události různé úrovně). Při testování mě samotného pak napadlo, že by možná nebylo na škodu při příští implementaci rozšířit aplikaci o nějaké okno, kde by se vypisovaly pouze události bez kalendáře a dalších přidanych událostí (například by se zde listovalo dny).

Na druhou stranu z testů vyplynulo, že náš návrh a implementace aplikace není vůbec špatná a že uživatel je schopen pochopit fungování aplikace relativně rychle.

8. Implementace GUI - nastavení mobilní aplikace [xkaras38]

Tato kapitola o implementaci se zabývá pouze částí mobilní aplikace, ačkoliv jsem se podílel i na desktopové aplikaci a dalších částech mobilní aplikace.

8.1 Implementace

Základem této části aplikace je okno Settings. To umožňuje dále přejít k oknu pro mazání dat a templatů. Toto okno obsahuje tlačítka pro smazání všech dat, templatů nebo obojího. Tlačítka jsou napojená přímo na funkce v modelu. Druhé tlačítko v settings nás posune do okna z templaty. To se skládá z kachliček pro každý template události. Kliknutím na jednotlivé z nich se dostaneme do detailu události upraveného pro templaty. Například se neobjevuje výběr data a času, protože ty nejsou u template potřebné. Dále nelze událost odebrat, jelikož jde o template a ten chceme aby šel pouze editovat. Změna dat je poté poslaná přes DataFlowViewModel do modelu. Následně můžeme tuto změnu pozorvat v oknu Add event na mobilu, kde nám již upravený template bude nabídnut.

8.2 Použité nástroje a knihovny

Vývoj této části probíhal za použití stejných knihoven a technologií jako v části zprávy o implementaci Davida Hudáka [xhudak03]. Pro rychlé shrnutí slouží následující odstavec.

Vývoj probíhal v rámci Visual Studia s technologiemi C#, Xamarin a Android Studia za použití NuGet knihovny pro zpracování JSON dat a kalendáře.

Velkou nevýhodou Xamarinu bylo jeho zprovoznění, kterou jsem musel vyřešit tím, že jsem použil jiný počítač. Obecně je tato technologie poměrně předimenzovaná pro někoho, kdo není s vývojem GUI ještě tolik sžitý. Podobné pocity jsem měl i s Android a Visual Studiem. Ty byli také náročné na výkon počítače. Pozitivem byl debugger a prediktivní doplňování, které usnadnilo práci např. u složitých názvů tříd Xamarinu.

8.3 Finální testování

Průběh testování byl distanční. Aplikace byla rozeslána přátelům, kteří byli ochotni věnovat mému projektu chvíli času. Těm poté bylo zadáno co mají zkusit udělat. Testování se nakonec povedlo provést se dvěma uživateli, u prvního se řešily především implementační chyby. Uživateli chyběla další nastavení jako například úprava vzhledu barev kachlíček templatů. Na druhou stranu ocenil možnost mazání samostatných templatů, jelikož si je zaplácal nesmysl a chtěl se vrátit do původního stavu. Dále ocenil koncept priorit, které mu umožní snáze se orientovat v nadcházejících událostech

Druhé testování proběhlo s pomocí sdílené obrazovky. Uživatel se v nastavení zorientoval poměrně rychle, avšak nejprve plně nepochopil funkcionalitu s přidáváním šablon (bylo nutné mu trochu napovědět). Uživatel u testování také prohodil, že by možná nebylo špatné přidat možnost odstraňovat jenom některého typu událostí (například s nízkou prioritou). Tedy že by v nastavení měl možnost odebrat všechny události dle nějaké kategorie. Také by uvítal, kdyby v nastavení šablon šlo nastavit nějaké opakování, například by byla zavedena událost narozeniny, která by se po přidání připsala do stejných dat v následujících letech. Na druhou stranu ocenil celý koncept šablon, který by mu pomohl zrychlení práce s kalendářem.

8.4 Vyhodnocení testu

Testování uživateli odhalilo, že návrh není tak intuitivní, jak jsme doufali. Toto by se dalo odstranit pomocí krátkého tutoriálu při prvním spuštění. Dále byli časté poznámky na chybějící funkcionality. Tomu by se dalo zabránit průběžným testováním s uživateli. Pozitivní ovšem byli ohlasy na koncept šablon a priorit. Z tohoto důvodu jsme si jistí, že poptávka po tomto typu kalendáře existuje a naše aplikace je krokem správným směrem. Příště bych se primárně zaměřil na proces tvorby. Konkrétně výše zmíněné průběžné testy s uživateli cílové skupiny.

9. Implementace GUI - desktopová aplikace [xosval04]

V této kapitole se budu zabývat tvorbou desktopové aplikace ve WPF.

9.1 Implementace

V tomto řešení aplikace bylo použito jedno okno a “druhé”, které vzniká překreslením a rozmazáním pozadí. V hlavním okně jsou položky rozmístěny do tří sloupců. V prvním se nachází kalendář a přehled aktivit na daný měsíc nebo den (dle přepnutí viz dále). V druhém je poté detail události. V něm se buď vytváří událost nová, nebo se upravuje událost vybraná z přehledu aktivit. Pro událost typu platba se po zaškrtnutí odpovídajícího checkboxu zobrazí další editovatelné položky pro cenu a měnu. Ve spodní části okna se může zobrazovat motivační citát v závislosti na nastavení. Poslední sloupec obsahuje tlačítka pro přidání, či odebrání události. Přepnutí mezi zobrazením událostí dne a měsíce a nakonec tlačítkem pro přepnutí do nastavení. Nastavení je výše zmíněné druhé okno. V něm je možnost nastavení řazení událostí v hlavním okně podle priority nebo data. Dále filtrace podle typu události a velikosti priority. Je zde i možnost pro nastavení zobrazení motivačních citátů. Nakonec je zde možnost pro smazání všech dat. Pro jednotlivé prvky byli použity nástroje toolboxu WPF a extended WPF toolkitu. Codebehind obsahuje hlavně logiku, při různých akcích jako jsou zmáčknutí tlačítek, či zapsání textu do textboxů a některé práce s těmito daty, které by byly jinak včleněny do viewModelu nepřírozně. Ve viewModelu jsou pak proměnná data, která vyžadují bindingy. Příkladem proměnné pro určení viditelnosti prvků. Mimo to jsou ve viewModelu i funkce které berou a zpracují data z modelu. V neposlední řadě zde jsou pomocné funkce pro práce s daty.

9.2 Použité nástroje a knihovny

Pro vývoj tohoto řešení byla použita technologie WPF v rámci Visual Studia s pomocí programovacího jazyka C#. Stejně jako u mobilní aplikace, i zde bylo využito NuGet rozšíření pro práci s JSON soubory, které byly využity v modelu aplikace. Dále bylo využito rozšíření WPF extended toolkit pro práci s volbou času, jelikož základní WPF tuto funkcionalitu neobsahuje. Použití tohoto nástroje se ukázalo jako nepříliš intuitivně ani designově vypadající, ale jako řešení zadávání času stačilo (lepší, než kdyby se muselo vše zadávat přes textová pole či slidery). Co se týká samotného WPF a Visual Studia, tak použití bylo poměrně jednoduché a narozdíl od problému s mobilní aplikací, bylo kompilování poměrně rychle a nenáročné. Osobně se mi líbil debugger ve Visual Studiu, který nabízí širokou škálu možností na vyhledávání chyb, například funkcionalitu watches, kam šlo zadat kteroukoliv proměnnou a okamžitě se vypisoval její aktuální obsah.

Osobně mi moc nevyhovovalo designování samotné aplikace z hlediska rozšiřování obrazovky s pomocí gridů s relativním pozicováním sloupců a řádků.

9.3 Finální testování

Testování proběhlo s kamarádem ze střední školy. Ten dostal exe soubor s naší aplikací a dostal za úkol vykonat základní činnosti, které nabízí naše uživatelské prostředí. První bylo přidat událost na správné datum. To zvládl správně. Další bylo přidat pár dalších událostí, což mu nedělalo problém, kromě situace, kdy omylem označil událost a nabídla se mu úprava události. Zde se totiž mírně mění prostředí a může to být pro uživatele matoucí. Se situací si naštěstí poradil. Dalším úkolem pak právě byla změna některých konkrétních vlastností u událostí. U druhé přidané události měl změnit pouze čas, u třetí události pak měl změnit vše včetně názvu. Další testy se nesly v podobném duchu. Dalším úkolem bylo v nastavení změnit výpis událostí pouze na ty, které jsou dané priority. To zvládl.

Následným zadáním bylo nastavení výpisu událostí pouze daného dne a událostí všech daného měsíce. Zde je použito tlačítko měnící svůj nápis podle toho, kam uživatel chce přejít. Bylo konstatováno, že to ze začátku není příliš intuitivní, protože nikde není zobrazen aktuální stav, nýbrž jen stav, na který se dá přejít.

V průběhu práce se mému kamarádovi zamlouvalo chytré filtrování pomocí priority i typu události. Možnost nastavit si motivační citáty. Velice nepříjemné ale bylo zadávání data, které je při prvním použití nepřírozené.

9.4 Vyhodnocení testu

Stejně jako u vyhodnocení Pilotního testu. Výsledky diskutovat, není nutná revize návrhu, stačí popis, co a jak je nutné do budoucna udělat jinak.

Uživatel zvládl testování poměrně dobře. Jediné, u čeho zaváhal, bylo odkliknutí vybrané události, jelikož na to bylo potřeba podržet ctrl a opět kliknout na vybranou událost. Bylo pro to přidáno tlačítko new. Dalším drobným problémem bylo pochopení výběru všech událostí daného měsíce a událostí pouze daného dne. Jinak se neobjevily nějaké větší problémy přímo s návrhem, jako spíše možná se škálou funkcionalit, které prostředí nabízí, takže do budoucna by se dalo přidat více možností, co lze dělat. Například přidat další možnosti řazení událostí, které by se například daly vybírat přímo u listu, větší práce s prioritami, nějaká možnost pro upozornění atd.

10. Závěr [xhudak03]

Výsledkem jsou funkční aplikace pro zápis událostí - jedna mobilní a jedna desktopová. Obě aplikace splňují svůj účel a fungují. Mobilní aplikace je pak zaměřena spíše na mladší publikum, které chce snadno a rychle zadávat činnosti do svého telefonu. Desktopová aplikace je pak zaměřena spíše na dospělé uživatele a nabízí spíše konzervativní vzhled.

Uživatelé obecně na mobilní aplikaci ocenili nápad přidávání snadných šablon, které v mnoha ohledech urychlují přidávání událostí, které jsou prostě nějakým způsobem generické a které se hodí zapsat hned (domácí úkol, test atp.) Jednoznačně se tedy jedná o přínos oproti ostatním aplikacím, kde šablony buď nejsou, nebo je jejich zadávání potlačené či

složitě. Co by uživatelé uvítali a co implementováno nebylo, pak bylo přidávání upozornění na události.

Na práci v týmu mi vyhovoval pohodový přístup kolegů a zkušenost s některými praktikami extrémního programování. V tomto typu programování jeden člen týmu psal kód a druhý (či i třetí člen) napovídal a pomáhal s tvorbou dané funkcionality. To se osvědčilo obzvlášť při debuggingu aplikace, kdy jsem neviděl zcela zjevnou chybu s prohozením názvů proměnných (místo `timeDay` jsem psal `dateTime`, což vedlo ke ztrátě dat psaním do jiné proměnné). Na práci v týmu se také osvědčila platforma `GitHub.com`, kde všichni členové měli okamžitě dostupné všechny vykonané změny v dílčích řešeních a mohli se tak okamžitě zapojit do vývoje. Přínosná také byla komunikační platforma `Discord`, kde jsme se vždy mohli snadno a rychle sejít a případně nasdílet obrazovku s naším problémem.

Co bych pak naopak udělal jinak, je lepší výběr tvůrčí platformy, jelikož `Visual Studio` jsem si během vývoje zcela neoblíbil a kompilování projektu na mobilu bylo extrémně dlouhé (prakticky tvořilo 60 - 70 % vývoje, což je opravdu příliš mnoho). Neuvěřitelným způsobem to brzdilo vývoj a omezilo to některé nápady, které jsem měl u implementace.

11. Závěr [xosval04]

Dosažený výsledek jsou 2 aplikace. Jedna mobilní, druhá pro počítač. Každá z nich pracuje se stejnými daty ale míří trochu jiným směrem. Mobilní se snaží o rychlost zadávání zatímco co počítačová cílí na rozlišné ovládací prvky.

Některé prvky které uživatelé kritizovali jsme na základě jejich hodnocení změnili. Jiné byly subjektivní nebo pevné kvůli implementaci. Uživatelé zejména ocenili možnost filtrování dle priorit či typu události. Tyto funkce se v jiných aplikacích nenachází.

Práce i komunikace v týmu fungovala na vynikající úrovni. Na všem jsme se rychle shodli a neproběhly žádné konflikty. Zejména mi vyhovovalo praktikování extrémního programování. Několikrát jsme využili toho že jeden člen mohl upozornit druhého na překlipy a malé chyby. Také se nám to osvědčilo při debugování chyb v aplikaci. Nebo při rozmýšlení implementace nových prvků.

Příště bych více zvážil jestli opravdu použít `Visual Studio` jelikož během vývoje bylo největším zdrojem frustrace i přes některé příjemné funkce.

12. Závěr [xkaras38]

Výsledným produktem je aplikaci na mobil i počítač. Každá má jiné klady a zápory. Mobilní aplikace nabízí prvky pro rychlejší práci (šablony). Ovšem práce s daty a jejich filtrací není plně dolazená. Na druhou stranu počítačová aplikace nabízí více možností pro filtrování událostí a práce v ní je o něco intuitivnější. Drobným negativem je neintuitivní zadávání data a času.

Uživatelé ocenili na mobilní aplikaci především koncept šablon. Který by využili pro většinu událostí. Dále je zaujal koncept priorit, který by usnadnil orientování ve větším množství dat. Jako negativum byla brána neintuitivnost rozhraní v některých oblastech aplikace.

U desktopové aplikace byla dobře hodnocená schopnost filtrování podle priorit a typu, která se u konkurence nenachází.

V týmu mi vyhovovalo, že jsme měli všichni podobný přístup i názory. Pokud ne, tak byla debata klidná a všichni byli otevření názorům druhých. To nám umožňovalo zkoušet praktiky extrémního programování. Například rychlé iterace měnící funkcionalitu výsledného produktu. Každý se podílel na návrhu architektury i na samotném řešení. Programování ve více lidech zároveň etc. V neposlední řadě, pokud měl kdokoliv problém ostatní mu jej ochotně pomohli řešit.

Příště bych více času věnoval výběru a nastavení prostředí. Pro člověka, který má rád minimalismus je základní verze VS příliš hektická.

Reference

Sem pište zejména důležité zdroje, odkud jste čerpali informace (knihy, články, časopisy, blogy apod.).

Odkazy na SW, manuály, fóra apod. uvádějte formou url do poznámky pod čarou.

<https://play.google.com/store/apps/details?id=com.google.android.calendar>

<https://play.google.com/store/apps/details?id=com.weekly.app>

<https://www.youtube.com/watch?v=QDmCL6hRjFw&list=PL4WEkbdagHIRv7ggbtrp969--y-goYS7d&index=14> (tutoriály, z nichž byla drtivá většina znalostí o vývoji v Xamarinu)