



Středoškolská technika 2022

Setkání a prezentace prací středoškolských studentů na ČVUT

MyDay - Android aplikace a web

Matěj Martan, Lukáš Fariz

**Střední škola informatiky a finančních služeb, Plzeň,
Klatovská tř. 2778/200G, 301 00 Plzeň**

Anotace

V naší práci jsme se zabývali problematikou psychických poruch a rozhodli jsme se vytvořit SOČ zaměřený na pomoc lidem trpícím panickou poruchou, depresí a úzkostí. Cílem projektu bylo vytvořit přehlednou a jednoduchou aplikaci a internetový portál, které poskytují ověřené informace a způsoby pomoci. Na vytvoření projektu byly použity jazyky Java, HTML, CSS a Javascript. Práce se zaměřila na vložení osobních zkušeností s psychickými poruchami spojených s informacemi od organizace WHO a překontrolovaných psychologem a psychiatrem. Výsledkem práce byla aplikace a internetový portál splňující kritéria přívětivého a přehledného prostředí pro lidi trpící zmíněnými problémy.

Klíčová slova

Psychické zdraví, aplikace, webový portál, deprese, úzkosti, panické ataky

Obsah

Úvod.....	4
Teoretická část	5
Motivace k vytvoření projektu.....	5
Smysl projektu	5
Popis obtíží, na které se aplikace zaměřuje.....	6
Deprese	6
Úzkost	6
Panická porucha	6
Porovnání s podobnými projekty	7
Nepanikař.....	7
VOS	8
Programovací jazyky	8
HTML	8
CSS	9
JavaScript.....	9
Důvody k výběru HTML, CSS a JS.....	9
Visual Studio Code	9
Adobe XD	10
Praktická část.....	11
Mobilní aplikace	11
Prvotní úvahy, výběr platformy	11
Programovací jazyk	11
Rozložení aplikace	12
Titulní strana	12
První krok identifikování při pomoci.....	14
Informační strana	16
Rozcestník aplikace při pomoci	17
První metoda pomoci	20
Druhá metoda pomoci.....	22
Třetí metoda pomoci	24
Čtvrtá metoda pomoci.....	26
Nastavení	27
Webová stránka.....	29
Prvotní úvahy, výběr platforem	29
Rozložení webové stránky	29
Úvodní stránka	29

Stránka zjistit více.....	29
Stránka stažení	29
Stránka Kontakty	30
Stránka Recenze.....	30
Responsivnost	30
Vysvětlení kódu	31
HTML	31
CSS	31
Závěr	33
Shrnutí.....	34
Použité zdroje informací	35

Úvod

V posledních letech se čím dál tím více setkáváme s lidmi, trpícími psychickými poruchami, či sami nějakou psychickou poruchou trpíme. Jde o skutečně velký nárůst, který úzce souvisí s moderní, uspěchanou dobou. Tato uspěchaná doba přispívá k vystavování tlaku na každého z nás. Tento tlak na každého působí jinak. V extrémních případech může docházet až k vyvolání nějaké psychické poruchy. V nejčastějších situacích jde právě o úzkost, která postiženou osobu tíží.

Z tohoto důvodu jsme se rozhodli pro projekt, který by obohacoval společnost a pomáhal v daném odvětví. Zároveň jsme také chtěli pomoci našemu blízkému okolí, či nám, které tyto problémy sužují.

Ze všeho nejdříve jsme si stanovili cíle, kterých bychom měli dosáhnout. U aplikace to byla hlavně přehlednost a jednoduchost, spojená s ověřenými způsoby pomoci v akutních situacích, či při pocitu, že by nějaká podobná situace mohla nastat. Tyto úkony měly být zpracovány formou čtyř aktivit pro odvedení pozornosti. U internetového portálu šlo především o informování potenciálního uživatele o stavu vývoje, spolu s možností stažení aplikace v podobě apk souboru, či možnosti zpětné vazby uživatelů. Cíle internetového portálu byly zhotoveny za využití komponentů, nebo funkcí Javascriptu.

Jelikož naše zkušenosti vycházely především ze tří psychických nemocí, jimiž byly: panická porucha, deprese, či úzkosti, tak tím pádem jsme se rozhodli pro zaměření na tyto tři konkrétní poruchy. Toto rozhodnutí nám umožnilo vložit do práce naše osobní poznatky s nestabilním mentálním zdravím spojené s ověřenými informacemi sdílenými organizací WHO a následovně překontrolovány psychologem a psychiatrem.

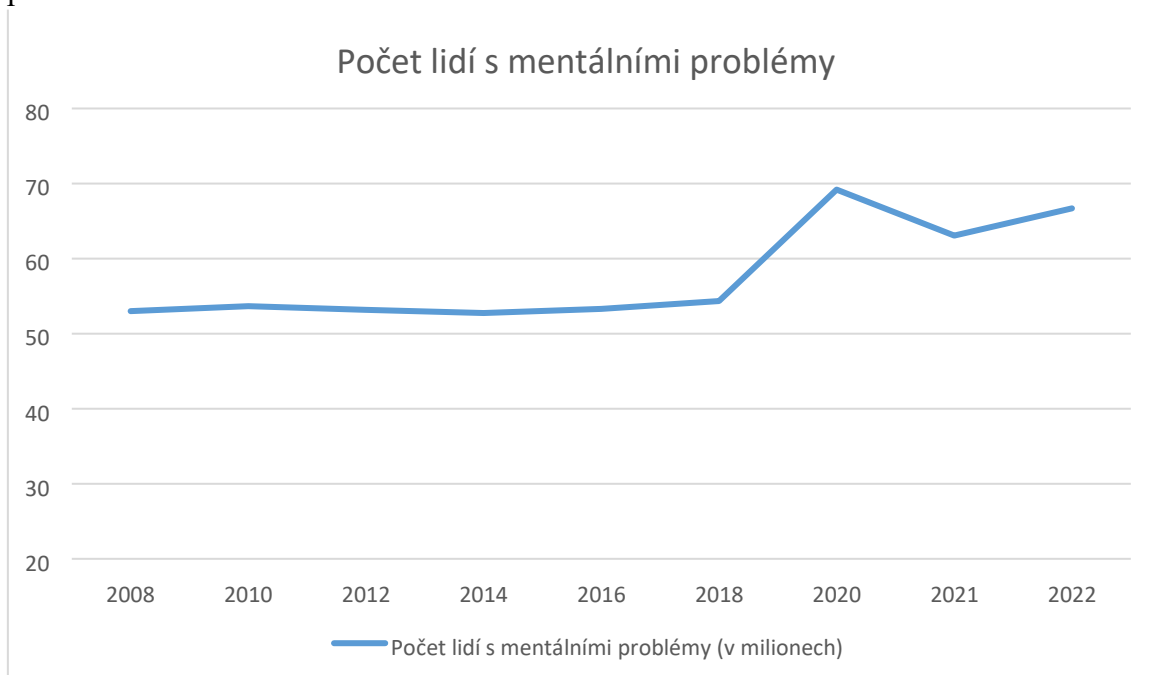
Po stanovení jednotlivých cílů bylo dalším krokem vyhledání informací k jednotlivým prostředkům, které budeme potřebovat k vypracování odborné práce. Díky tomuto jsme zjistili základní informace ke každému jednotlivému pracovnímu postupu a oba jsme si určili vhodný programovací jazyk. K android aplikaci se jím stala Java, ve vývojovém prostředí Android Studio. U internetového portálu se stal vhodným jazykem HTML, spojen se základy CSS a Javascriptu.

Chtěli bychom vytvořit přehledné prostředí pro lidi, jež trpí zmíněnými problémy, které by mělo na uživatele působit přátelsky. Proto bychom chtěli vytvořit aplikaci, kterou si zájemci bezplatně stáhnout na webovém portále a dozvědět se zde doplňující informace ohledně dané problematiky. Aplikace i webový portál splňují výše zmíněná kritéria.

Teoretická část

Motivace k vytvoření projektu

Jak jsme již uvedli v úvodu, přivedla nás osobní zkušenost k nápadu vytvořit aplikaci na dané téma. Vnímali jsme to tak, že tím můžeme pomoci dalším lidem s podobnými obtížemi. Na projektu jsme začali pracovat koncem léta roku 2022. Z veřejně dostupných zdrojů, ať již jsou to mainstreamová média či odborná periodika, bylo možné v tomto období zjistit, že skokově narůstá počet mladých lidí a dětí s obdobnými problémy, jako se potýkáme my. Stav nárůstu lze vidět na grafu viz. Graf 1, který zahrnuje celkový počet lidí s mentálními problémy v USA. Tento stav nastává i v Evropě, tudíž je zřejmé, že se jedná o globální problém.



Graf 1 Statistika lidí s mentálními problémy v USA (zdroj: 6; 7; 8)

Smysl projektu

Jevy, které jsme popsali v prvním odstavci, se z velké části týkají mladých osob. Generace, která již žije v souladu s mobilními technologiemi, které jsou zároveň jejich primární zdroj informací. Proto jsme neuvažovali o tištěném letáku, inzerátu a podobně, ale o jednoduché aplikaci, šířené zdarma prostřednictvím internetu tak, aby si ji každý, který vlastní android telefon, mohl nainstalovat.

Vidíme v aplikaci prostředek, který by mladým lidem s problémy, jež jsme popisovali, nejprve ukázal, že nejsou osamoceni, že nejsou divní, že naopak jsou jedni z mnoha, kdo se s tímto potýká. Kdo by jim ukázal, že jejich problémům rozumí a dokáže se do jejich pocitů ponořit. A v konečném důsledku, kdo dokáže ukázat cestu a pomoci. Naším cílem není nahrazovat odbornou lékařskou pomoc. Smyslem je pomoci lidem se v jejich obtížích zorientovat a v případě, jsou-li jejich problémy akutní, je k lékařské pomoci navést. Aplikace má snahu naučit uživatele lépe své problémy identifikovat a svému blízkému okolí popsat.

Popis obtíží, na které se aplikace zaměřuje

Deprese

Jedná se o jednu z nejrozšířenějších psychických poruch. V celosvětovém měřítku až 3.8% populace trpí depresemi a jejich počet stále narůstá. Reakce na tento duševní stav bývá u jednotlivců různá. Od snížení schopnosti prožívat radost, až po myšlenky na sebevraždu. Z toho důvodu je deprese velmi nebezpečná porucha, protože může být i smrtelná, v případě, že není aktivně řešena. Častěji se deprese vyskytuje u žen než u mužů, dle světové zdravotní organizace WHO.

Projevuje se různými psychickými příznaky. Příkladem může být pokles energie, aktivity a zájmů, snížení schopnosti radovat se či možnosti se koncentrovat. Také dochází ke snadné unavitelnosti, poklesu sebevědomí, poruchám spánku a v neposlední řadě k úzkostem či psychomotorickému neklidu. Právě při takovém úzkostném stavu je vhodné využít jednu z forem pomoci, jež aplikace nabízí. [1;5]

Dopad však může mít i na fyziologii člověka a projevovat se například bolestmi hlavy, problémy s ostrým viděním, pocitem nedostatku vzduchu, či dušení, pocitem vnitřního třesu nebo hyperventilací.

Možným řešením deprese je biologická léčba nebo psychoterapie. Léčbu je nutné započít co možná nejdříve a v rozsahu, dle potřeby pacienta.

Úzkost

Úzce souvisí s depresí. Úzkost cítí během života každý z nás v menším měřítku. Je spojena částečně se stresem, např. před závažným životním rozhodnutím. Jejím spouštěčem může být také složitá kombinace emocí, zahrnující strach, zlé předtuchy a obavy. V některých případech může být úzkost i dlouhodobá. Její intenzita se může lišit od lehkého neklidu až po stav paniky.

Jejími psychickými příznaky jsou kupříkladu neklid, psychické napětí či lekavost. Právě u těchto příznaků je vhodné využít aplikací nabízenou formu pomoci. Hodící se způsob je nutné uvážit dle závažnosti příznaků. Ovšem nejčastěji vyhovujícím způsobem bývá provedení dechových cviků podle aplikace. [2;4;5]

Fyzické příznaky mohou být zejména bušení srdce, pocit nevolnosti, bolest na hrudi, zkrácené dýchání, chvění končetin a v některých případech až neschopnost ovládat konkrétní svaly.

Možné řešení úzkosti záleží na intenzitě a času trvání. Léčba může probíhat dvěma způsoby, ale ve většině případů se využívá obou. Prvním způsobem je psychoterapie, která je často podpořena biologickou léčbou, jež je druhým způsobem. V praxi léčba probíhá konzultací s terapeutem, který s pacientem diskutuje o problémech a obavách. Případně, je-li intenzita úzkosti silná, doporučí, co dělat v akutních situacích.

Panická porucha

Jedná se o poruchu, při které dochází k opakujícím se extrémním pocitům úzkosti. Těmito situacím se říká panická ataka. Nemají pevný řád, kdy se objevují, často je to ve chvíli relativní duševní pohody, tj. vyvoláno vnitřním neklidem člověka. V některých případech

ovšem člověk nemusí znát příčinu tohoto stavu, zda jaká myšlenka jej spustila. Trvají 5 až 30 minut ve většině případech. Člověk si během nich představuje katastrofické scénáře z různých podnětů, zvýší se mu tep a zrychlí dýchání. Jedná se o epizodu vnitřního neklidu, úzkosti a strachu.

Panická porucha se vyznačuje například nadměrným stresem, neklidem, psychickým vypětím, lekavostí, pocitem ohrožení, pocit depersonalizace, či derealizace nebo pocitem ztráty kontroly nad vlastním tělem. U výše zmíněných příznaků je silně doporučeno využít metod, které aplikace nabízí. Metody jsou koncipovány na tyto situace a jsou schopné v brzkém stádiu nadcházející ataky a příznaky zmírnit, až potlačit. [3;4;5]

Depersonalizace je stav člověka, kdy si připadá, jako by byl divákem nějakého filmu a jen přihlížel. Pacient se v některých případech nemůže hýbat a pouze mu proudí myšlenky hlavou. Připadá si, že sleduje sebe i život ve 3. osobě. Při derealizaci ovšem vnímá svět, jako by to byla nějaká videohra, či film, při kterém je hlavním protagonistou. Jevy kolem něj mu neprijdou za žádnou cenu reálné a lidé kolem něj se mu mohou jevit až nebezpeční. Někdy u ostatních lidí může vidět tzv. auru. Aurou je myšlené záření kolem dané osoby.

Často při panické atace dochází k doprovodným fyzickým projevům jako je kupříkladu nadměrné pocení, zvýšená tepová frekvence, silné bušení srdce, tlak na hrudi, návaly horka či chladu, zrychlený dech, dušnost, bolesti hlavy, břicha, nevolnost, sucho v ústech, sevřené hrdlo či zvýšený krevní tlak.

Při první panické atace může mít člověk pocit, že se jedná o infarkt. Je to pro něj absolutně nový zážitek a často z tohoto důvodu volá záchrannou službu. Většinou do příjezdu záchranné služby tento stav opadne. Těmto stavům lze předejít za použití různých úkonů, které pomáhají se stresem.

Léčba probíhá podobně jako u úzkosti. Tudiž dochází k psychoterapii a nasazení biologické léčby. Terapeut se snaží na sezeních najít příčinu panických atak a obeznámí pacienta s pomáhajícími postupy při akutních stavech.

Porovnání s podobnými projekty

Nepanikař

Nepanikař je mobilní aplikace pro Android a iOS, která se zaměřuje na pomoc lidem trpícím úzkostí, stresem a podobnými duševními problémy. Aplikace obsahuje řadu nástrojů a funkcí, které pomáhají uživatelům s jejich mentálním zdravím a umožňují jim lépe se vyrovnat s každodenními stresy.

Mezi hlavní funkce aplikace patří relaxační cvičení, která pomáhají uživatelům uklidnit mysl a zmírnit napětí. Aplikace také obsahuje seznam rad a tipů pro zlepšení mentálního zdraví, jako je dostatek spánku, pravidelné cvičení a správná výživa. Kromě toho aplikace Nepanikař umožňuje uživatelům zaznamenávat své pocity a myšlenky pomocí deníkové funkce a sledovat jejich progres v práci na svém mentálním zdraví. Aplikace také nabízí možnost konzultace s odborníkem prostřednictvím online chatu nebo e-mailu pro další podporu a rady.

V závěru lze říci, že aplikace Neapanikař je užitečný nástroj pro lidi trpící úzkostí a stresem, který poskytuje řadu nástrojů a funkcí pro podporu mentálního zdraví a lepší vyrovnavání se s každodenními stresy.

V porovnání lze říci, že obě stránky jsou zaměřené na poskytování pomoci lidem s psychickými poruchami a informací o této problematice. Oba projekty jsou zcela zdarma, ale liší se převážně ve zpracování.

VOS

Aplikace umožňuje uživatelům zaznamenávat své myšlenky, pocity a úkoly a sledovat jejich progres v práci na svém mentálním zdraví. Mezi hlavní funkce aplikace patří možnost zaznamenávat pocity, nálady a úkoly pomocí deníkové funkce, a také možnost přidávat fotografie a další multimédia. Aplikace také obsahuje řadu relaxačních cvičení a rad pro podporu mentálního zdraví, jako je dostatek spánku, pravidelné cvičení a správná výživa.

Kromě toho aplikace VOS, umožňuje uživatelům spolupracovat s ostatními pomocí sdílení odkazů nebo tlačítek "sdílet" a také poskytuje možnost konzultace s odborníkem prostřednictvím online chatu nebo e-mailu pro další podporu a rady.

Aplikace VOS: Wellbeing & Intelligent Journal je užitečný nástroj pro podporu mentálního zdraví a lepší vyrovnavání se s každodenními stresy. Umožňuje uživatelům zaznamenávat a sledovat své myšlenky, pocity a úkoly a poskytuje řadu nástrojů a rad pro podporu mentálního zdraví. Hlavní rozdíl mezi MyDay a Vos je kompletně zdarma, zatímco na využití všech funkcí Vos si musí uživatel zaplatit. Jejich webový portál je také zaměřen spíše ke komerčním účelům (prodej oblečení a NFT).

Programovací jazyky

HTML

HTML (HyperText Markup Language) je základním jazykem webových stránek. Používá se ke strukturování obsahu na webových stránkách a k popisu vzhledu stránek pomocí předdefinovaných značek. HTML sám o sobě neumožňuje vytvoření složitějších grafických prvků ani interaktivity, ale spolu s jazyky jako CSS a JavaScript umožňuje vytvářet moderní a atraktivní webové stránky.

V současnosti se používá nejnovější verze HTML5, která přináší mnoho nových funkcí a možností. Například umožňuje lepší integraci s multimédií, umožňuje vytvoření interaktivních map a zlepšuje možnosti pro použití na mobilních zařízeních.

HTML se skládá ze značek, které jsou umístěny mezi hranatými závorkami a označují různé prvky obsahu, jako jsou odstavce, nadpisy, seznamy nebo odkazy. Značky se navzájem spojují a tvoří tak strukturu stránky. Díky tomu mohou prohlížeče zobrazit obsah stránky správně a uživatelé se snadno orientovat na stránce.

HTML je základem každé webové stránky a je velmi důležité, aby byl správně použit. Správně napsaný HTML zajišťuje, že stránka bude správně zobrazena na všech prohlížečích a zařízeních a bude nabízet informace přehledně.

CSS

CSS (Cascading Style Sheets) je jazyk, který slouží k formátování webových stránek. Používá se k definování vzhledu prvků na stránce, jako jsou barvy, velikosti písma, pozadí nebo umístění prvků. CSS umožňuje vytvářet atraktivní a přehledné stránky bez nutnosti ručního upravování HTML kódu. CSS se používá spolu s HTML nebo jinými jazyky pro tvorbu webových stránek. HTML sám o sobě umožňuje pouze základní formátování obsahu, ale CSS umožňuje vytvořit složitější a přesnější vzhled stránek. Například můžete pomocí CSS definovat, jak budou vypadat odkazy na stránce nebo jak budou zobrazeny tabulky.

CSS je velmi důležité pro tvorbu moderních a atraktivních webových stránek. Umožňuje snadné měnění vzhledu stránek bez nutnosti ručního upravování HTML kódu. Díky CSS je možné vytvářet stránky, které jsou příjemné pro oči a snadno se s nimi pracuje.

JavaScript

JavaScript je jedním z nejpoužívanějších programovacích jazyků na webu. Jeho hlavním účelem je přidávat dynamiku a interaktivitu do webových stránek. Jeho vývoj se datuje do roku 1995, kdy ho vytvořil Netscape Communications Corporation jako způsob, jak rozšířit možnosti webových prohlížečů za účelem vytvoření bohatšího uživatelského prostředí.

JavaScript se používá hlavně pro vytvoření skriptů, které se spouštějí na straně klienta. To znamená, že skripty se spouštějí na počítači uživatele, nikoli na serveru, což má za následek rychlejší odezvu aplikace. JavaScript se také často používá pro vytvoření animací, zobrazování aktuálního času a datumu, validaci formulářů a mnoho dalších úloh.

JavaScript je také velmi flexibilní a lze ho použít spolu s jinými technologiemi, jako je například HTML a CSS, pro vytvoření pokročilých webových aplikací. Díky své široké podpoře ve většině webových prohlížečů se stal JavaScript jednou z nejdůležitějších technologií pro vývoj webu.

Důvody k výběru HTML, CSS a JS

Hlavním důvodem výběru těchto jazyků, byla zkušenost s těmito jazyky z prvního ročníku. Preferoval jsem tyto jazyky oproti např. PHP, se kterým jsem zatím nepochoval. Vývojové prostředí HTML, CSS a JS je jednoduché na úpravu a design pomocí CSS může být pestrý a moderní.

Visual Studio Code

Visual Studio Code je populární textový editor pro programování a vývoj softwaru, který je vyvinut společností Microsoft

Jednou z hlavních výhod Visual Studio Code je jeho podpora pro mnoho jazyků programování, včetně C++, C#, Java, Python, JavaScript a mnoha dalších. K dispozici jsou také různé rozšíření, které umožňují přidávat další funkce a nástroje pro specifické potřeby a účely. Například rozšíření "Python" umožňuje přidat podporu pro Python a integrovat nástroje jako debugger a linter. Visual Studio Code také nabízí řadu nástrojů pro vylepšení vývoje a práce s kódem. Například integrovaný debugger umožňuje ladit kód, zatímco IntelliSense poskytuje kontextové návrhy kódu a pomáhá

s psaním kódu. K dispozici je také integrovaný terminál, který umožňuje přímý přístup k příkazové řádce a umožňuje spouštět příkazy přímo z editoru.

Kromě toho Visual Studio Code podporuje práci v týmech a umožňuje spolupráci na projektech pomocí funkcí jako sdílení kódu a spolupráce na dokumentech. Je také kompatibilní s mnoha populárními verzovacími systémy, jako je Git, a umožňuje integraci s různými nástroji pro správu kódu a projektů.

V závěru lze říci, že Visual Studio Code je vysoce výkonný a flexibilní nástroj pro vývoj softwaru, který poskytuje širokou škálu funkcí a možností pro usnadnění práce.

Adobe XD

Na design webu byl použit Adobe XD. Je nástroj pro navrhování a prototypování uživatelských rozhraní, který je specificky navržený pro tvorbu webových a mobilních aplikací. Adobe XD umožňuje snadno navrhnout a prototypovat webové stránky a aplikace s použitím přímočarých nástrojů pro návrh a s použitím předdefinovaných prvků a předloh. Adobe XD umožňuje snadno spolupracovat s ostatními členy týmu přes sdílení prototypů nebo společné práce v reálném čase. Adobe XD obsahuje mnoho užitečných nástrojů pro testování a zlepšování designu, včetně nástrojů pro vytváření uživatelských testů a pro sledování používání.

Další výhodou Adobe XD je integrace v Creative Cloud od společnosti Adobe, což umožňuje snadno exportovat návrhy do dalších aplikací pro tvorbu webových stránek, jako je například Adobe Photoshop nebo Adobe Illustrator.

Praktická část

Mobilní aplikace

Prvotní úvahy, výběr platformy

Po výběru tématu nás čekalo těžké rozhodování, jímž bylo, jak mobilní aplikaci koncipovat. Dalo by se říci, že jsme měli na výběr celkově ze tří návrhů, které by bylo možné provést. Nejdříve jsme si ovšem museli určit na kterou mobilní platformu hodláme cílit. Jelikož jsme především zastánci Android zařízení, tak jsem se rozhodl právě pro tuto platformu. Tomuto rozhodnutí také napomáhala jednoduchost šíření aplikace na cizí zařízení, jež je oproti iOS platformě celkem značná. Tímto verdiktem se mé dosavadní návrhy zmenšily na počet dvou konceptů.

Prvním konceptem bylo využití open-source frameworku React Native. Ten běží na JavaScript „threadech“. V dnešních dnech už především na nadstavbě JavaScriptu, jímž je TypeScript. Výhodou React Nativu by při zpracování projektu bylo hned několik. Hlavní výhodou by bylo, že aplikace by byla dostupná, jak pro platformu Android, tak pro platformu iOS. Ovšem tato výhoda by šla na úkor rychlosti aplikace. Zároveň jsme s tímto frameworkem neměli tolik zkušeností, proto jsem se rozhodl od tohoto konceptu oprostít.

Druhým, zároveň i zbývajícím konceptem, který by naplňoval naše kritéria, bylo využití Android Studia. Jak už z názvu vypovídá, tak se jedná o vývojové prostředí, které se zaměřuje na mobilní vývoj Android aplikací pro mobilní zařízení. Samotné Android Studio je vlastněno firmou Google, jež je také vlastníkem samotného Androidu a bylo představeno 16. května 2013. Je založeno na již běžícím IDE IntelliJ IDEA. Jeho hlavní výhodou je svižnost, které může aplikace dosáhnout a jakého designu je aplikace schopná docílit. Tyto dva faktory byly rozhodující, že i přes větší náročnost vývoje, jsme se rozhodli zvolit právě jej.

Programovací jazyk

Jak už jsme zmínili při výběru platformy, tak naše rozhodnutí padlo na využití vývojového prostředí Android Studio. V samotném prostředí lze vyvíjet za pomoci jednoho ze dvou dostupných jazyků. Prvním je Java, zatímco druhým je Kotlin. Jelikož s Kotlinem bohužel nemáme žádné zkušenosti, tak jsme se rozhodli pro výběr Javy.

Co je Java vůbec za programovací jazyk? Je to objektivě orientovaný programovací jazyk, jež je v dnešní době značně rozšířený. Běží téměř na pěti a půl miliardách zařízení po celém světě a je každodenně využíván více než šesti milióny vývojářů. Je vlastněn firmou Oracle Corporation, která se zaměřuje na designování, prodej, výrobu softwaru a hardwaru.

Syntaxe je značně zjednodušena oproti jazykům C a C++. Chybí zde některé low-level prvky programovacích jazyků, jako jsou pointery, preprocesor a další. To je pro Javu výhodou, avšak také nevýhodou. Omezuje se nám tímto využitelnost u vybraných úkolů, kde bychom tyto funkce využili. Naopak to ale zjednodušuje úkony, při práci s pamětí. Právě zde využijeme výhodu Javy oproti C, kterou je Garbage Collection. Tuto výhodu využijeme v situacích, kdy například přestaneme využívat nějaký objekt a nebudeme k němu mít žádnou referenci. V tento moment ji tzv. garbage collector najede na heap paměti a následně vymaže.[9]

Další výhodou Javy je to, že je interpretovaný. To znamená, že se nevytváří zdrojový kód, ale pouze bytecode. Díky tomuto můžeme takový kód spustit na libovolném zařízení, které má k dispozici Java Virtual Machine neboli JVM. Toto je možné díky tomu, že formát bytecode je nezávislý na architektuře počítače.

Rozložení aplikace

Samotná aplikace je tvořena z několika částí neboli screenů. Tyto části by se daly rozdělit do dvou kategorií. Počínaje kategorií, jež má za úkol uživatele navést k různým formám pomoci, či ho

o problematice informovat, konče kategorií, která je spíše doplňková a není spojená s částí pomoci uživatelům.

Titulní strana

Titulní strana je zpracována tak, aby uživatele zaujala. Aby jeho prvotní dojem byl co nejlepší a nepůsobila zbytečně komplikovaně. Je to takové těžiště aplikace. Uživateli nabízí výběr, do jaké části aplikace se potřebuje dostat. Je tvořena sadou tlačítek, které uživatele navigují do vybraných oblastí.

Velkou většinu ze sady tlačítek tvoří tlačítka „citová“. Citová z toho důvodu, že vyjadřují pocit, jak se uživatel může v danou chvíli cítit. Jsou zbarvena do celkově tří barev, kde každá z barev vyjadřuje pocit jistoty, štěstí a relaxace.

Zbylou část ze sady tlačítek jsou tlačítka „doplňková“. Je jím ovšem jediný zástupce, a to je tlačítko Settings. Doplňková jsou z důvodu toho, že uživatele nenavigují na hlavní funkci aplikace, kterou je pomoc v akutních případech nebo vzdělání o problematice, ale navigují uživatele do screenu s nastavením.

Kód aplikace má poměrně jednoduchou strukturu viz Obrázek 1. Každé jednotlivé tlačítko si inicializujeme a přiřadíme mu jeho unikátní id, kterým ho rozlišíme od ostatních. Toto vše probíhá v metodě onCreate z důvodu toho, že kód, který je zde napsaný se bude vykonávat při načítání prvotním načítání screenu. Potom každému tlačítku přiřadíme naslouchač neboli listener, ve kterém pomocí vnitřní anonymní třídy zavoláme potřebnou metodu a nastavíme globální proměnnou „clicked“ dle id tlačítka. Globální proměnná „clicked“ je zde pro budoucí vývoj aplikace, k dnešnímu dni není plně využívána. U citových tlačítek spouštíme metodu „openMomentalniProblem“, která uživatele přesouvá na další krok při potřebné pomoci. Zatím co u doplňkových tlačítek spouštíme metodu „openSettingsActivity“, jež uživatele naviguje do screenu s nastavením.

```

/**
 * Domovskú screen aplikace
 */
public class MainActivity extends AppCompatActivity {
    public Button button1,button2,button3,button4,button5,button6,button7;
    public static int clicked;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getSupportActionBar().hide();
        setContentView(R.layout.activity_main);

        /**
         * Listener, pro všechna tlačítka
         */
        TextView settings = (TextView) findViewById(R.id.settingsID);
        settings.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openSettingsActivity();
            }
        });

        button1 = (Button) findViewById(R.id.button); //tlačí mě na hrudi
        button1.setOnClickListener(new View.OnClickListener(){
            public void onClick(View view){
                openMomentalniProblem();
                clicked = 1;
            }
        });
        button2 = (Button) findViewById(R.id.button2); //jsem k ničemu
        button2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openMomentalniProblem();
                clicked = 2;
            }
        });
    }
}

```

Obrázek 1 Ukázka kódu pro inicializaci a přiřazení tlačítek (zdroj: autor)



Obrázek 2 Aktuální snímek obrazovky titulní strany (zdroj: autor)

První krok identifikování při pomoci

Smyslem tohoto screenu je identifikování, zda je nutná akutní pomoc. Uživatele vybízí k výběru, zda má momentálně problém, či nikoliv.

V případě, že se problém vyskytuje, tak uživatele přesměruje na screen, který nabízí různé možnosti pomoci, nebo ho dokáže telefonicky spojit s linkou první psychické pomoci.

Pokud ale uživatel zvolí, že teď nemá akutní problém, tak ho aplikace nasměruje k části, kde si může přečíst o problematice bližší informace. To se hodí například pokud měl uživatel dříve problém a ví, jak se u tohoto problému cítil, tak tyto informace může využít. V aplikaci vybere, jak se problém projevoval, či jak se cítil, zvolí, že se nejedná o akutní problém a zde na něj vyskočí informace, jak se deprese, úzkost či panická porucha projevují. Dále se také dozví, jak se v takovém případě léčí a různé příznaky, které nepříjemným stavům předcházejí.

Z hlediska kódového zpracování je screen koncipován poměrně jednoduše. Prvotně je nutné inicializovat obě tlačítka. Následně přepíšeme implementaci, tzv. override, metody „onBackPressed“ a dosadíme do ní naši vlastní metodu „navigateBack“, viz Obrázek 3, která uživatele přesouvá na domovskou obrazovku. Nyní v metodě „onCreate“ přiřadíme inicializovaným tlačítkům jejich id a následně nastavíme na každé tlačítko zvlášť naslouchač neboli „OnClickListener“, ve kterém pomocí vnitřní anonymní třídy spustíme námi potřebnou metodu.

U tlačítka „Ano“ je jím metoda „openAktivita2“, která má za úkol navigovat na Aktivitu2, jež je rozcestím pro výběr pomoci. U tlačítka „Ne“ jím bude metoda „openCoToVystihuje“, která uživatele přesouvá na screen s vypsáním informací o problematice a jejich různými vlastnostmi.

```

public class momentalniProblem extends AppCompatActivity {
    protected Button button8,button9;//tlacitka 8 - ano tlacitka 9 - ne
    static int clicked = MainActivity.clicked;

    /**
     * Navigátor zpět na domovskou screen
     */
    public void navigateBack(){
        Intent intent = new Intent( packageContext this, MainActivity.class);
        startActivity(intent);
    }
    /**
     * OVERRIDUJE SE Z DUVODU ZJEDNODUSSENI. UZIVATEL NEMUSI MACKAT XKRAT BACK TLACITKO JAKO NEGRAMOT
     * **/
    @Override
    public void onBackPressed(){
        navigateBack();
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getSupportActionBar().hide();
        setContentView(R.layout.activity_momentalni_problem);
        Button button8 = findViewById(R.id.button8);
        Button button9 = findViewById(R.id.button9);

        /**
         * Listener, který rozhoduje o stisknutém tlačítku
         */
        button8.setOnClickListener(new View.OnClickListener() { //tlacitko ANO
            @Override
            public void onClick(View v) {
                Log.i( tag: "problemANO",Integer.toString(clicked));
                openAktivita2();
            }
        }
    }
}

```

Obrázek 3 Ukázka kódu, který se stará o navigaci na domovskou stránku, inicializuje tlačítka, přiřazuje id a spouští příčné metody (zdroj: autor)



Obrázek 4 Ukázka dosavadního vzhledu stránky, jež rozhoduje, zda je nutná akutní pomoc, či pouze informování o problematice (zdroj: autor)

Informační strana

Její podstatou je uživatele informovat o základních informacích každých zmíněných poruch. Případně jak se projevují, či jejich motivy. K zobrazení této strany je důležité stisknout tlačítko „Ne“ na dotazové straně, jež se ptá, zda má dotyčný momentální problém.

Screen je složen z několika komponentů. V horní třetině obrazovky je zobrazen nadpis, pod kterým se ve zbývajících dvou třetinách zobrazuje tzv. ScrollView viz Obrázek 6. Tento ScrollView je takový box neboli kontejner, ve kterém se zobrazují ostatní komponenty podle námi zvolených kritérií. Tímto mám na mysli, že si můžeme vybrat, jak budou komponenty rozloženy v tomto kontejneru. Já v aktuálním případě zvolil „LinearLayout“, jež je orientovaný vertikálně. To znamená, že komponenty budou řazeny pod sebe.

V případě kódového zpracování je tento screen odlišný od ostatních. V javovské třídě tohoto souboru najdeme pouze metodu „onCreate“, která je povinná pro všechny screeny, která zdědí všechny potřebné prvky a přiřadíme rozložení pro celou obrazovku. Zároveň také skryjeme záhlaví aplikace, jímž je „actionBar“. Zbytek magie se děje v xml souboru screenu, který je designem aplikace. Celý kód je zde tvořen XML tagy s doprovodnými parametry viz Obrázek 5.

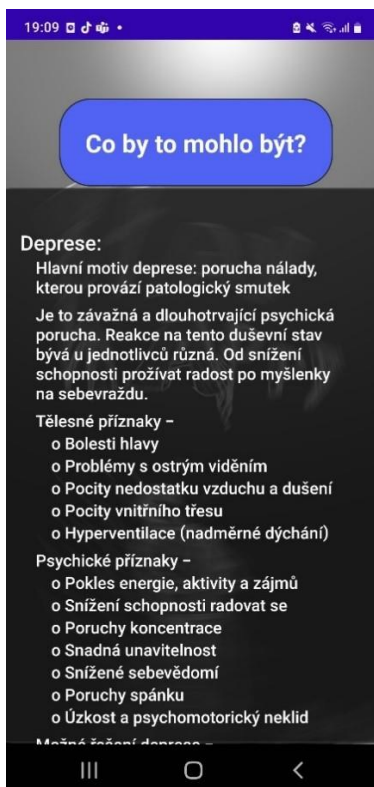
```
<TextView
    android:id="@+id/textView21"
    android:layout_width="411dp"
    android:layout_height="538dp"
    android:width="300dp"
    android:height="300dp"
    android:background="@drawable/boxvystihuje"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/View2" />

<ScrollView
    android:id="@+id/scroll"
    android:layout_width="373dp"
    android:layout_height="525dp"
    android:layout_marginStart="15dp"
    android:layout_marginTop="179dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="12dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="SpeakableTextPresentCheck">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="20dp"
        android:layout_marginRight="20dp"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView22"
            android:layout_width="wrap_content"
```

Obrázek 5 XML kód, jež nastavuje nadpis a ScrollView (zdroj: autor)



Obrázek 6 Momentální vzhled informační strany (zdroj: autor)

Rozcestník aplikace při pomoci

Jde o pomyslný rozcestník aplikace, na kterém si uživatel zvolí ze 4 typů pomoci. K zobrazení této strany je nutné při identifikaci k pomoci zvolit možnost „Ano“, jež znamená, že se jedná

o momentální problém. Jeho smyslem je dát uživateli na výběr a navést ho k použití některé z uvedených typů.

Celá obrazovka je složena z více částí. V horní části nalezneme dvě tlačítka, informaci, k jakému účelu screen slouží a text, jež má za úkol uklidnit uživatele. První tlačítko je určené k navigaci na domovskou obrazovku aplikace. Druhé tlačítko slouží při velmi akutních případech. Při stisku tohoto tlačítka se uživateli zobrazí dialogové okno, zda si přeje být spojen s linkou první psychické pomoci. V tento moment může uživatel případné zavolání zrušit, anebo ho naopak potvrdit. V případě, že uživatel potvrdí telefonát je automaticky vytáčeno telefonní číslo linky první psychické pomoci. Ve střední části aplikace se nacházejí 4 velká bílá tlačítka viz Obrázek 8. Každé tlačítko signalizuje jednotlivou metodu pomoci, která se nabízí. Zároveň tlačítka obsahují ilustrace, spolu s popisy každé aktivity, která se za nimi skrývá.

Kód, který zpracovává logiku za tímto screenem je jedním z nejsložitějších z celé aplikace. Důvodem toho je, že se zde musíme dotazovat na permise k telefonování, kontrolovat jejich stav a zpracovávat dialog. To je zde vyřešeno pomocí několika metod. Hlavní z nich je privátní metoda „hasPermission“, která v sobě ukrývá jednu podmínku, viz Obrázek 7, jež se dotazuje, zda má daný screen permise pro použití vytáčení telefonu, neboli jsou nastaveny na hodnotu 1. Pokud ano, tak jen vypíše do konzole, že jsou již povolené a nic dále neprovádí.

Ovšem v případě, že tyto práva nemá, tak je spuštěna metoda „requestCallPermissions“, jež se dotáže přes pomocníka „ActivityCompat“ na povolení permisí. Pro uživatele je to pouhé okno, které se zobrazí při prvotním využití možnosti telefonátu na linku první psychické pomoci. Celá tato metoda se spouští v jiné metodě „openPhoneApp“, která zobrazuje dialog, který se ptá uživatele, zda chce opravdu zavolat na zmíněnou linku. Metoda funguje na principu, že se nejdříve dotáže, zda má uživatel permise přes metodu „hasPermission“, pak za využití třídy „AlertDialog“ vytvoří objekt. Tomuto objektu nastavíme titulek, zprávu na co se uživatele ptáme a dvě tlačítka. Prvním tlačítko nastavíme tak, že na našem objektu použijeme metodu „setPositiveButton“, kde v prvním parametru nastavíme jméno a v druhém vytvoříme vnitřní anonymní třídu ve které přepíšeme implementaci metody „onClick“. V této metodě vytvoříme objekt „intent“ a zároveň mu nastavíme parametry pro vytáčení telefonu a převedenou uri telefonního čísla dané linky. Pak už jen použijeme metodu „startActivity“, kde dáme jako parametr náš objekt „intent“ a máme první tlačítko hotovo. Druhé tlačítko bude o poznání jednodušší. Na objektu použijeme metodu „setNegativeButton“, které v prvním parametru nastavíme text tlačítka na „Ne“ a místo vnitřní anonymní třídy použijeme null. Teď už jen zobrazíme objekt pomocí metody „show“ a metoda „openPhoneApp“ je hotová. Nyní vytvoříme metody, které budou přesouvat uživatele na konkrétní postupy pomoci, vytvoříme objekty tlačítek a přiřadíme jim přes id hodnoty. Na každém tlačítku pak použijeme „listener“, který přes vnitřní anonymní třídu bude mít u každého tlačítka přiřazenou potřebnou metodu pro přesun, nebo zobrazení dialogu. Posledním krokem u tohoto screenu je už jen přepsání implementace metody „onBackPressed“, abychom si zaručili to, že tlačítko zpět vždy bude navigovat na screen s dotazem, zda se jedná o momentální problém.

```

/**
 * Metoda volající request na permission pro telefonování
 */
private void hasPermission(){
    if(ContextCompat.checkSelfPermission( context: Aktivita2.this,
        Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED){
        Log.i( tag: "PERMISSION", msg: "HAVE ALREADY");
    }
    else{
        requestCallPermissions();
    }
}

/**
 * Pomocná metoda pro request permission pro telefonování
 */
private void requestCallPermissions(){
    ActivityCompat.requestPermissions( activity: this, new String[] {Manifest.permission.CALL_PHONE}, CALL_PERMISSION_CODE);
}

/**
 * Metoda, která spouští dialog okna. Po potvrzení volá linku první psychické pomoci
 */
public void openPhoneApp(){
    hasPermission();

    AlertDialog.Builder question = new AlertDialog.Builder( context: this);
    question.create();
    question.setTitle("Chceš si promluvit s nějakým odborníkem?");
    question.setMessage("Chceš si promluvit s nějakým odborníkem? Kliknutím ANO budeš telefonicky spojen.");
    //Vyvolává telefonický hovor s novou linkou
    question.setPositiveButton( text: "ANO", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Intent intent = new Intent(Intent.ACTION_CALL,Uri.parse("tel:116123"));
            startActivity(intent);
        }
    });
    question.setNegativeButton( text: "NE", listener: null);
    question.show();
}

```

Obrázek 7 Kódové zpracování permisí a dialogu pro dotaz uživatele (zdroj: autor)



Obrázek 8 Vzhled rozcestníku (zdroj: autor)

První metoda pomoci

K první z metod „Vyjmenuj co vidíš a slyšíš“ se dostaneme při zmáčknutí stejnojmenného tlačítka na daném rozcestníku. Postup nabádá uživatele k vypsání 5 věcí, které kolem sebe vidí či slyší. Toto řešení pomáhá dotyčnému oprostít se od jeho myšlenek, které ho v ten moment mohou napadat, jež jsou vesměs negativní, úzkostné až depresivní. Právě tato metoda se snaží ukázat dotyčnému krásu světa kolem něj. Po ukončení aktivity je uživatel dotázán, zda byl postup účinný. V případě, že ano má možnost z aktivity odejít, nebo si ji znovu zopakovat. V případě, že ne, má na výběr ze tří možností. První možností je zopakovat cvičení a dát mu druhou šanci. Druhou možností je vyzkoušet jiné cvičení, při jejímž výběru je uživatel přesunut na další metodu. Třetí metodou je opustit aktivitu a vrátit se zpět na rozcestí, kde uživatel vybere metodu dle svého výběru.

Z hlediska kódového zpracování je screen podobný předchozímu. Začneme tím, že si vytvoříme objekty tlačítek, textových inputů a vytvoříme arraylist. Každému objektu přiřadíme pomocí id hodnotu. Pak na potvrzovacím tlačítku opět vytvoříme listener, který bude využívat vnitřní anonymní třídy. Zde při každém zmáčknutí tlačítka nastavíme to, aby se do arraylistu přidal text z každého inputu a spustíme metodu „showDialog“. Tuto metodu si vytvoříme podobně, jako při vytváření dialogu u rozcestníku. Nastavíme důležité parametry typu: titulek, zprávu, první a druhé tlačítko. Následně ji pak zobrazíme. U prvního tlačítka budeme spouštět naši vlastní metodu „showDialogBetter“, viz Obrázek 9, jež vytváří podobný dialog, tentokrát s upraveným textem a ptá se jak uživatel hodlá pokračovat.

V případě, že chce opakovat cvičení, tak vymaže z inputů text a nechá ho pouze v arraylistu. U druhého tlačítka budeme také spouštět vlastní metodu, která se nazývá „showDialogWorse“, funguje také na stejném principu při tvorbě dialogu, ovšem taky s jiným titulkem, zprávou a funkcí tlačítek. Tentokrát tu jsou tři tlačítka. První umožňuje uživateli odejít ze screenu na rozcestník aplikace. Pokud ho zmáčkne, tak zavolá metodu „onBackPressed“ a je přesunut. Druhé tlačítko uživateli nabízí možnost vyzkoušet jiný postup pomoci na jiném screenu. Uživatel u něj zavolá metodu „transferNext“, která ho přesune na další postup za pomocí objektu intent. Třetí tlačítko navrhuje vyzkoušet postup znovu. Při jeho zmáčknutí se vymaže text v inputu. Nyní vytvoříme metodu „control“. Ta má za úkol kontrolovat kolik aktivit uživatel využil a případně se ho dotázat, zda chce pokračovat pomocí alertdialogu. Funguje na stejném principu jako ostatní metody s dialogy. Je zde pouze pozměněn titulek, zpráva a obě tlačítka. První tlačítko je, že chce uživatel pokračovat a má listener null. Druhé tlačítko přes vnitřní anonymní třídu volá metodu „HandleGoHome“, která uživatele vrací přes objekt intent na titulní stranu aplikace v případě, že už nechce pokračovat.

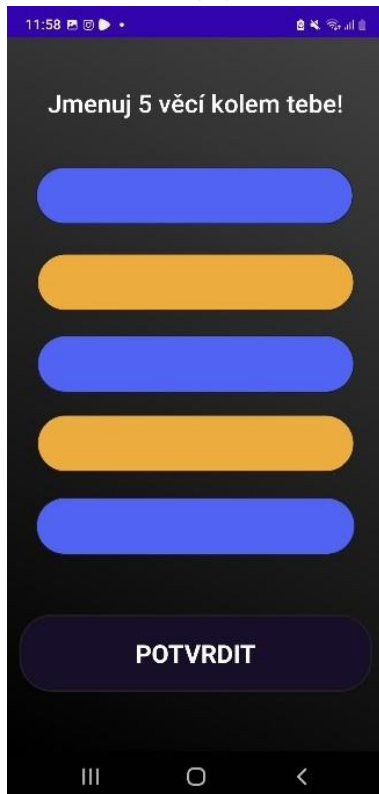
```

/**
 * Metoda, zobrazující dialog, zda se uživatel cítí lépe a reaguje na odpovědi
 */
public void showDialog(){
    AlertDialog.Builder dialog = new AlertDialog.Builder( context: this);
    dialog.create();
    dialog.setTitle("MyMind");
    dialog.setMessage("Cítíš se lépe?");
    dialog.setPositiveButton( text: "Lépe", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) { showDialogBetter(); }
    });
    dialog.setNegativeButton( text: "Stále ne", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) { showDialogWorse(); }
    });
    dialog.show();
}

/**
 * Pomocná metoda pro showDialog. Zobrazuje další dialog, který se zobrazí v případě, že se uživatel cítí lépe a zvolil tuto možnost
 */
public void showDialogBetter(){
    AlertDialog.Builder dialog = new AlertDialog.Builder( context: this);
    dialog.create();
    dialog.setMessage("To jsme moc rádi :)");
    dialog.setNeutralButton( text: "Ok", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            onBackPressed();
        }
    });
    dialog.setPositiveButton( text: "Zkusit znovu", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            input1.setText("");
            input2.setText("");
            input3.setText("");
            input4.setText("");
            input5.setText("");
        }
    });
}

```

Obrázek 9 Kód, který vytváří 2 ze 3 dialogů (zdroj: autor)



Obrázek 10 Vzhled prvního postupu pomoci (zdroj: autor)

Druhá metoda pomoci

Na druhý postup pomoci se dostaneme zmáčknutím tlačítka „Soustřed' se na jiné objekty“ u rozcestníku aplikace. Tato aktivita se dotazuje uživatele na 5 různých otázek. Otázky jsou koncipované tak, aby uživatel přemýšlel nad nějakým jeho vysněným místem, oblíbenou vzpomínkou, viz Obrázek 11, či jeho oblíbenou aktivitou. Poslední otázkou je to, jak se uživatel po uvědomění těchto věcí cítí. Toto uživatelské uvědomění mu ukazuje krásy života, na co se může těšit a to, že život stojí za to žít. Po dokončení je uživatel opět dotázán, zda mu aktivita pomohla

a má na výběr stejné možnosti, jak pro ukončení aktivity, přesunu na další metodu, či opětovné opakování postupu.

Kód je tvořen podobnou strukturou jako předchozí aktivita. Začínáme tím, že vytvoříme `arraylist`, číselnou proměnnou `„count“` spolu s objekty textových `importů` a tlačítek. Těmto pomocí `id` přiřadíme hodnoty a na tlačítko potvrzování nastavíme `listener`. Tento `listener` opět pomocí vnitřní anonymní třídy bude obsahovat metodu `„onClick“`, která bude tvořena jedním velkým `switchem`. `Switch` se bude řídit pomocí proměnné `count`, která se po každém zmáčknutí zvětší o 1. Každý `case` ve `switchi` mění text obsažený v titulku podle následované otázky, přidává text z `inputu` do `arraylistu` a maže text v `inputu` viz Obrázek 11. Jakmile dojde k poslední otázce, tak se proměnná `resetuje`

a spustí se metoda `„showDialogDone“`. Tato metoda znovu obsahuje vytvoření dialogu a jeho nastavení důležitých parametrů viz. `titulek`, `zpráva` a dvě tlačítka. První tlačítko vyvolává metodu `„showDialogAgain“` a druhé metodu `„showDialogWorse“`. Obě tyto metody vytváří další dialog, kterému nastavují stejné vlastnosti, jako předchozímu dialogu. Ovšem `„showDialogWorse“` obsahuje tři tlačítka. První pro možnost opakování aktivity.

Druhé pro přesunutí na jiné cvičení. Třetí k odchodu z aktivity na rozcestník. K těmto funkcím jsou pomocné metody: `„HandleMove“`

a `„handleGoChoose“`. `„HandleMove“` slouží k přesunu uživatele. `„handleGoChoose“` má za účel přesunout uživatele na rozcestník. Nyní doplníme do metody `„onCreate“`, kde také přiřazujeme objektům tlačítek `id`, metodu `„control“`. Ta má jako v předešlém případě za úkol kontrolovat kolikrát uživatel spustil určitou aktivitu a po dosažení určitého počtu zobrazí vlastní dialog zda chce pokračovat. V případě, že ano, tak `listener` na tlačítku u dialogu bude `null`. Pokud už nechce pokračovat, tak vytvoříme znovu vnitřní anonymní třídu s metodou `„onClick“` a spustíme metodu `„HandleGoHome“`, která pomocí internetu naviguje na titulní stranu aplikace.

```
text = (EditText) findViewById(R.id.input20);
Title = (EditText) findViewById(R.id.Title);

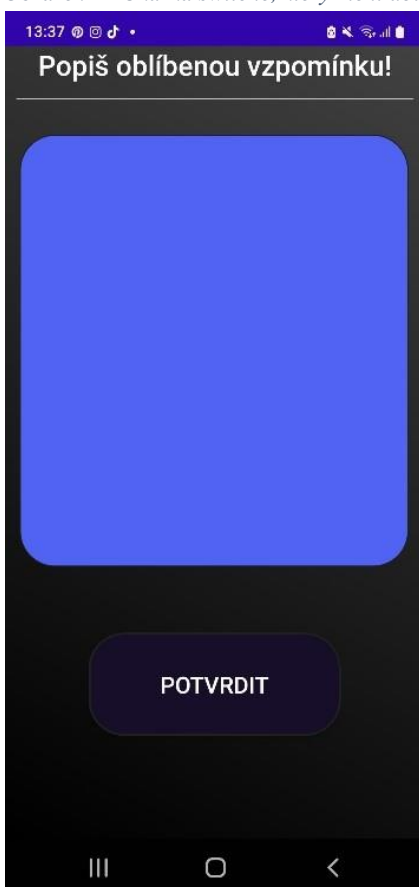
Aktivita2.i++;
control();

System.out.println("Aktivita7 " + Aktivita2.i);

/**
 * Event listener, měnící názvy a ukládající data do listu podle počtu stisknutí tlačítka
 */
button = (Button) findViewById(R.id.button17);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        count++;
        Log.i("tag: "countCase: ", Integer.toString(count));
        switch (count) {
            case 1:
                Title.setText("Popiš objekt vedle tebe!");
                popisData.add(text.getText());
                text.setText("");
                break;
            case 2:
                Title.setText("Popiš vysněné místo!");
                popisData.add(text.getText());
                text.setText("");
                break;
            case 3:
                Title.setText("Popiš oblíbenou vzpomínku!");
                popisData.add(text.getText());
                text.setText("");
                break;
            case 4:

```

Obrázek 11 Ukázka switche, který mění dotazy při stisknutí tlačítka (zdroj: autor)



Obrázek 12 Ukázka jedné z otázek při aktivitě (zdroj: autor)

Třetí metoda pomoci

Ke třetí metodě pomoci se lze dostat v rozcestníku po zmáčknutí tlačítka „Kontrolované dýchání“. Jak už z názvu vypovídá, tak se bude jednat o různé techniky dýchání, které mají za úkol uživatele zklidnit. Je zde na výběr ze tří technik, které jsou doporučovány psychology, či psychiatry a mnou osobně vyzkoušeny. První technika trvá nejkratší dobu, zatím co třetí trvá dvojnásobek doby oproti první technice. Ke každé metodě jsou zobrazeny ilustrace a počítání času viz Obrázek 14.

Tato metoda je z hlediska kódu jedna ze složitějších. Je to z důvodu, že se zde vyskytují fragmenty, které se jednotlivě mění po stisku tlačítka. V kódu nejprve začneme vytvořením objektů tlačítek

a přiřadíme pomocí id hodnoty. Následně na každém z nich spustíme listener a v něm vnitřní anonymní třídu s metodou „onClick“. V této metodě na objektu „Toast“ zavoláme metodu „makeText“, která nám zobrazí po zmáčknutí tlačítka notificační okno s námi nastaveným textem. Následně vytvoříme objekt FragmentManageru, který získáme z getteru „getSupportFragmentManager“. Na něm pak zavoláme „beginTransaction“ s doplňujícími parametry. Prvním z nich bude „replace“, kde nastavíme parametry id „fragmentContainerView“, název našeho fragmentu a null argumenty. Druhým je pak „setReorderingAllowed“ nastavený na true. To nám zajistí, že animace budou fungovat správně a budou tak optimalizované. Třetím je „addToBackStack“, což povoluje, aby uživatel mohl vyskočit zpět z fragmentu. Čtvrtým je už jen commit, což pošle tento konkrétní fragment, aby byl viditelný. Tyto kroky zopakujeme u všech tlačítek, ovšem s jiným id a také fragmentem. Dále vytvoříme metody jako jsou: control, transferNext, showDialog, showDialogBetter, showDialogWorse, handleGoHome. Všechny tyto metody jsou identické jako u předchozí metody pomoci, tudíž je využijeme. Nyní nám stačí přepsat metodu „onBackPressed“ a zavolat zde naši vlastní metodu „navigateBack“, která uživatele, pomocí intentu, přesouvá na rozcestník aplikace.

Nyní vytvoříme fragment ve kterém vytvoříme objekt tlačítka, zobrazování obrázku a objekty textových polí. Následně v metodě „onCreateView“ přiřadíme objektům jejich hodnoty a vytvoříme další objekt typu „View“ s id layoutu našeho fragmentu. Na tlačítku nastavíme listener opět s metodou „onClick“. V té nastavíme hodnoty textových polí, které mají zobrazovat, také první instrukci procesu a nastavíme je všechny viditelné na našem layoutu. Teď vytvoříme nový CountdownTimer, jež bude mít v parametrech jak dlouho má počítat a po kolika milisekundách. Zde vytvoříme proměnnou po kolika vteřinách budeme chtít počítat, v našem případě 5, a nyní uděláme metody: onTick, onFinish. V metodě „onTick“ vytvoříme string „sDuration“ a „finalDuration“. „sDuration“ bude obsahovat hodnotu z naší proměnné a „finalDuration“ hodnotu z parametru metody. Naši proměnnou zmenšíme o 1000 milisekund a vytvoříme switch, který se bude řídit podle stringu „sDuration“ viz Obrázek 13. Pokud hodnota bude 15 nebo 5, tak změním text dle instrukcí, změním obrázek a znovu resetujeme naši proměnnou. Pokud hodnota bude 10, tak provedeme stejné úkony, ovšem s jiným obrázkem a textem. Po ukončení switche nastavíme obsah textového pole na string „finalDuration“. V metodě „onFinish“ nastavíme viditelnost všech nepotřebných objektů na gone a opět změním text na dokončeno. Zároveň také zobrazíme dialog metodou „showDialog“ z třídy třetí metody pomoci. Tyto úkony musíme provést u každého stylu dýchání. Jistě s jinými obrázky, časy a texty.


```

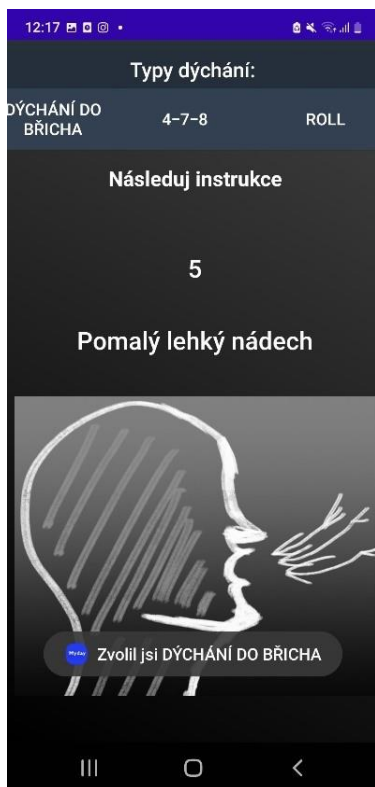
title.setText("Následuj instrukce");
counter.setVisibility(screen.VISIBLE);
image.setVisibility(screen.VISIBLE);
image.setImageResource(R.drawable.inhale);
helpText.setText("Pomalý lehký nádech");
startProcess.setVisibility(screen.GONE);

new CountdownTimer(durationStart, countdownInterval: 1000) {
    long separateCounter = 5000; /** POČÍTÁ PO 5 VTEŘINÁCH **/
    @Override
    public void onTick(Long millisUntilFinished) {
        String sDuration = Long.toString(TimeUnit.MILLISECONDS.toSeconds(millisUntilFinished));
        String finalDuration = Long.toString(TimeUnit.MILLISECONDS.toSeconds(separateCounter));
        separateCounter = separateCounter - 1000;
        switch (sDuration) {
            case "15" :
            case "5" :
                helpText.setText("Pomalý lehký výdech");
                image.setImageResource(R.drawable.exhale);
                separateCounter = 5000;
                break;
            case "10" :
                helpText.setText("Pomalý lehký nádech");
                image.setImageResource(R.drawable.inhale);
                separateCounter = 5000;
                break;
        }
        counter.setText(finalDuration);
    }

    @Override
    public void onFinish() {
        counter.setVisibility(screen.GONE);
        image.setVisibility(screen.GONE);
        startProcess.setVisibility(screen.VISIBLE);
        title.setText("Čas je to zkoušet znovu?");
        helpText.setText("Skvělá práce");
        ((Aktivita4) getActivity()).showDialog(); // ZOBRAZUJE DIALOG PO DOKONČENÍ
    }
}.start();

```

Obrázek 13 Ukázka stopek z fragmentu (zdroj: autor)



Obrázek 14 Ukázka 1. stylu dýchání s Toastem (zdroj: autor)

Čtvrtá metoda pomoci

K poslední metodě pomoci je možné se dostat z rozcestníku přes stisk tlačítka „Citáty, které tě nakopnou“. Je tvořena formou citátů viz Obrázek 16. Jsou zde vybrány citáty, které mají člověku ukázat jeho krásu a smysluplnost.

V kódu si vytvoříme opět objekty textových polí, tlačítek. Dále také dva arraylisty pro autory a citáty spolu s číslem, indikátorem, aktuálního citátu. Následně přiřadíme objektům hodnoty z id

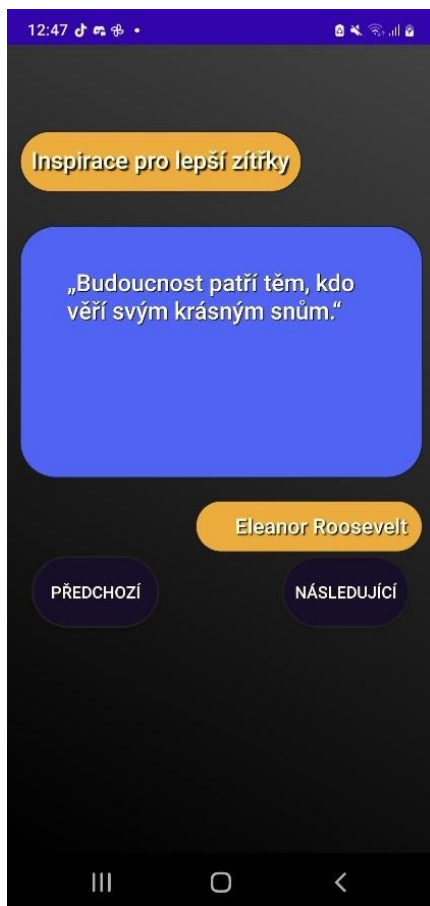
a indikátor nastavíme na 0. Nyní v try bloku vytvoříme objekt `BufferedReader`, který má jako parametr `InputStreamReader`, jež ze složky `Assets` získává soubor „Citáty.txt“. Vytvoříme pole stringů a pomocný string. Do pomocného stringu načteme první řádek ze souboru a uděláme while cyklus, který poběží dokud pomocný string nebude null viz Obrázek 15. Ve stringu nastavíme pole jako splitnutý pomocný string. Pak přidáme do arraylistu citátů z pole na indexu 0 a do arraylistu autorů z pole indexu 1. Po ukončení cyklu zavřeme `BufferedReader` a zavoláme metodu „`setBothText`“, která nastaví textové pole autora a citátu z arraylistu na pozici indikátoru. V catch bloku nastavíme textové pole na omluvu a autorův text na prázdný text. Ošetříme všechny potřebné výjimky jako jsou: `FileNotFoundException`, `IOException`. Nyní nastavíme listenery na obě tlačítka. V první vnitřní anonymní třídě nastavíme jednoduchou podmínku, kterou bude, že indikátor bude větší nebo rovno 1, následně zmenšíme indikátor o 1 a zavoláme metodu „`setBothText`“. V druhé vnitřní anonymní třídě bude také podmínka, kterou bude, že indikátor bude menší než délka arraylistu – 1 a následně zvětšíme indikátor o 1 a také zavoláme metodu „`setBothText`“. Nyní uděláme metody „`control`“ a „`HandleGoHome`“, které jsou identické jako v předchozí metodě pomoci. Tudíž je využijeme.

```
indicator = 0; //ukazatel na jakém jsme indexu v listu citátů a autorů

/**
 * Načítání citátů ze souboru
 */
try {
    //Načítá z assets složky txt file... FileReader nefunguje
    BufferedReader bufferedReader = new BufferedReader(
        new InputStreamReader(getAssets().open( fileName: "Citaty.txt")));

    String corrector[];
    helper = bufferedReader.readLine();
    while(helper != null){
        corrector = helper.split( regex: ";" );
        quotesList.add(corrector[0]);
        authorList.add(corrector[1]);
        helper = bufferedReader.readLine();
    }
    bufferedReader.close();
    setBothText();
} catch (FileNotFoundException e404){
    mainText.setText("Omlouváme se, citáty momentálně nejsou k dispozici...");
    authorText.setText("");
    System.out.println(e404);
}
catch (IOException e) {
    mainText.setText("Omlouváme se, citáty momentálně nejsou k dispozici...");
    authorText.setText("");
    e.printStackTrace();
}
before.setOnClickListeners(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(indicator>=1){
            indicator--;
            setBothText();
        }
    }
});
after.setOnClickListeners(new View.OnClickListener() {
```

Obrázek 15 Ukázka kódu načítání ze souboru (zdroj: autor)



Obrázek 16 Ukázka aktivity v aplikaci (zdroj: autor)

Nastavení

Nyní jsme se dostali k tlačítkům „doplňkovým“. Tyto tlačítka tvoří pouze různé extra funkce, které nejsou nutné pro hlavní funkčnost aplikace. Jedním z nich je i tlačítko nastavení viz Obrázek 18. Po jeho stisku se dostaneme na screen s nastavením, kde si můžeme rozkliknout Instagram MyDay nebo se dostat na stránku MyDay.

Z hlediska kódu se nejedná o složitou záležitost. Prvně si vytvoříme objekty tlačítka a metodu „openUrl“. Ta bude mít jako parametr string url. V těle metody bude nový objekt „uri“ typem Uri, který bude tvořen metodou „parse“ ze třídy uri a jako parametr metody bude naše „url“ viz Obrázek 17. Nyní vytvoříme intent, který bude mít jako parametr „Intent.ACTION_VIEW“ a druhým naše uri. Pak už jen zavoláme „startActivity“ s naším intentem. V metodě „onCreate“ přiřadíme hodnoty objektům tlačítka pomocí id a na každý dáme listener s vnitřní anonymní třídou s metodou „onClick“. V metodě budeme spouštět naši vlastní metodu „openUrl“ s požadovaným url. V prvním případě to bude odkaz na Instagram a ve druhém odkaz na web. Tento listener zopakujeme ještě jednou u druhého tlačítka a přiřadíme druhému odkaz.

```

private void openUrl(String url){
    Uri uri = Uri.parse(url);
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
    startActivity(intent);
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_settings);
    setTitle(Html.fromHtml( source: "<b>Nastavení</b>"));
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    ColorDrawable colorDrawable = new ColorDrawable(Color.parseColor( colorString: "#080808"

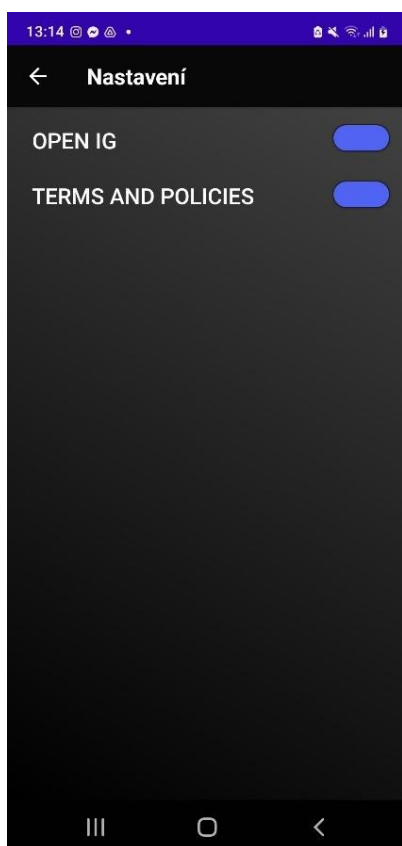
    getSupportActionBar().setBackgroundDrawable(colorDrawable);

    button10=findViewById(R.id.button10);
    button11=findViewById(R.id.button11);

    button10.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            openUrl("https://www.instagram.com/myday.mentalhealth.app/");
        }
    });
    button11.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { openUrl("https://mydayandroid.tode.cz/"); }
    });
}

```

Obrázek 17 Ukázka kódu nastavení (zdroj: autor)



Obrázek 18 Ukázka aktivity v aplikaci (zdroj: autor)

Webová stránka

Prvotní úvahy, výběr platform

Při vytváření aplikace jsme došli k závěru, že je nutné rozšířit povědomí o aplikaci, informovat o funkcích aplikace a dát možnost jejího následného stažení. Právě všechny tyto věci splňuje webová stránka, kterou jsme se rozhodli vytvořit. Webová stránka je zpracovaná převážně ve Visual Studio Code. K prvotnímu editování byl použit Adobe Dreamweaver, ale později byl web editován pouze přes Visual Studio Code.

Rozložení webové stránky

Webová stránka je rozdělena na několik dalších stránek, které jsou prokliknutelné z úvodní stránky.

Úvodní stránka

Když se uživatel dostane na webovou stránku, tak je důležité, aby se na stránce dokázal orientovat. Proto je úvodní stránka zpracována jednoduchým moderním designem, který zároveň uživatele zaujme. Jako první informace, které si už na toho všimne je základní popis principu a funkčnosti aplikace a webu. Ten se nachází na úvodní stránce. Spolu s ním zde nalezneme odkazy na sociální sítě, které nás informují o aktuálním stavu aplikace a mnoho dalších věcech. Youtube kanál nabízí následné ukázky aplikace pomocí nahrávané obrazovky přímo z telefonu. Tím uživatel vidí přesně jak aplikace funguje. Instagramový profil se soustředí na rozšíření povědomí o existenci aplikace a poutavou grafikou se snaží zaujmout co největší publikum. Discord server slouží jako poradna a uživatel zde nás na prostor na položení jakýchkoliv případných dotazů ohledně aplikace či webu. Na discord server se může připojit kdokoliv, pomocí odkazů na webové stránce. Přístup na tyto sociální sítě nalezne uživatel přehledně na úvodní stránce.

Stránka zjistit více

Další důležitý prvek na stránce, který slouží na získání dalších informací je tlačítko zjistit více. Ta uživatele odkáže na další stránku, na které si uživatel vybere mezi jednotlivými poruchami. Podle vybrané poruchy je přesměrován na text, který náleží k vybrané poruše. Text informuje, co daná porucha je jaké má příznaky tělesné a psychické. Následně se dozví uživatel u možných řešeních vybrané poruchy.

Jak bylo již dříve zmíněno a polykat co se soustředí na 3 poruchy panickou ataku, depresi a úzkost. Tyto poruchy jsou dost rozdílné a každá vyžaduje unikátní řešení. Abychom poskytli uživatelům efektivní a ověřenou pomoc, zajistili jsme, aby všechny informace použité v aplikaci i na webu byly ověřené psychologem. mimo to většina informací vychází z webových stránek významných světových organizací WHO. Vyhodnotili jsme, že tento zdroj je nejvíce důvěryhodný a ověřený. Úkol těchto informací zmíněných na webu je spíše pro informaci.

Stránka stažení

Jako hlavní činitel pomoci slouží aplikace. Ta je na webu přímo dostupná ke stažení. Je tak učiněno kartou stáhni zde. po kliknutí na toto tlačítko umístěné v menu webové stránky se aplikace automaticky stáhne a uživatel si následně může přímo nainstalovat.

Stránka Kontakty

Na této stránce se nachází odkazy na sociální sítě, které nás informují o aktuálním stavu aplikace a mnoho dalších věcech. Youtube kanál nabízí následné ukázky aplikace pomocí nahrávané obrazovky přímo z telefonu. Tím uživatel vidí přesně jak aplikace funguje. Instagramový profil se soustředí na rozšíření povědomí o existenci aplikace a poutavou grafikou se snaží zaujmout co největší publikum. Discord server slouží jako poradna a uživatel zde nás na prostor na položení jakýchkoliv případných dotazů ohledně aplikace či webu. Na discord server se může připojit kdokoliv, pomocí odkazů na webové stránce. Přístup na tyto sociální sítě nalezne uživatel přehledně na úvodní stránce.

Stránka Recenze

Tato stránka poskytuje uživateli možnost zpětné vazby. Slouží k vytvoření webové stránky se zobrazením hvězdičkového formuláře pro hodnocení aplikace MyDay.

Ve značce <body> se nachází obsah stránky, který je rozdělen do různých divů s třídami "container", "post", "text", "edit", "star-widget" a "textarea". První div třídy "container" slouží jako obal pro celou stránku. Div třídy "post" zobrazuje text "Děkujeme za Vaše hodnocení" a tlačítko "Upravit recenzi", které slouží k opětovnému zobrazení hvězdičkového formuláře po odeslání hodnocení.

Hvězdičkový formulář je vytvořen pomocí radiobuttonů a štítků s ikonami písma (pomocí třídy "fas fa-star"). Formulář také obsahuje textarea pro zadání textové recenze a tlačítko pro odeslání hodnocení.

Nakonec se nachází JavaScript kód, který se stará o zobrazení a skrytí hvězdičkového formuláře a textu po odeslání hodnocení. Když uživatel klikne na tlačítko „Poslat“, hvězdičkový formulář se skryje a zobrazí se text "Děkujeme za Vaše hodnocení" a tlačítko "Upravit recenzi". Kliknutím na tlačítko „Upravit recenzi" se opět zobrazí hvězdičkový formulář a skryje se text a tlačítko.

Jedná se tedy o hvězdičkové hodnocení aplikace MyDay a umožňuje uživateli odeslat svou recenzi a případně popsat svoji osobní zkušenost s aplikací. Po odeslání je možné hodnocení upravit pomocí tlačítka „Upravit recenzi“. Stránka je navíc formátována pomocí externích stylů a ikony jsou zobrazeny pomocí font-awesome knihovny.

Responsivnost

Tato webová stránka je responsivní, což znamená, že se dokáže přizpůsobit různým velikostem obrazovky, ať už se jedná o počítač, tablet nebo mobilní telefon. To je zajištěno pomocí metatagu s názvem "viewport", který se nachází ve značce <head>. Tento metatag umožňuje webovému prohlížeči přizpůsobit velikost stránky na základě šířky zařízení, na kterém je stránka zobrazována.

Dalším znakem responsivnosti této stránky je použití relativních jednotek místo absolutních jednotek, například použití procentuálních hodnot místo pevných hodnot v pixelech pro šířku a výšku různých prvků. To umožňuje prvkům přizpůsobit se různým velikostem obrazovky.

Vysvětlení kódu

HTML

V hlavičce stránky je nastavený dokumentový typ (DOCTYPE) na HTML a je také zadán jazyk (`lang="cs"`), což znamená, že stránka je v češtině. V hlavičce také můžeme vidět několik meta tagů, které slouží k nastavení různých vlastností stránky, jako je kódování znaků (`charset`), kompatibilita s prohlížeči (`http-equiv`) a velikost zobrazení (`viewport`). Dále v hlavičce najdeme název stránky (`title`) a odkazy na externí styly a fonty (`link`). Tělo stránky (`body`) se skládá z různých elementů, jako je navigační menu (`nav`), hlavní sekce (`main`) a další prvky obsahující text, obrázky a odkazy. Stránka také obsahuje prvky pro mobilní zařízení, jako je rozevírací menu a vyhledávací pole. Všechny tyto prvky jsou navzájem propojené pomocí odkazů a stylů, díky čemuž vzniká celkový vzhled a funkčnost stránky.

Dalším zajímavým prvkem v kódu HTML je element `div` s třídou `"search-input-container"`, který obsahuje vstupní pole (`input`) pro vyhledávání. Tento prvek slouží k tomu, aby uživatel mohl na stránce vyhledávat obsah, který je pro něj relevantní.

Navigační menu je tvořeno elementem `nav` a obsahuje odkazy na různé sekce stránky a další důležité informace. Hlavní sekce (`main`) obsahuje text a obrázek, které slouží k představení aplikace MyDay a jejích funkcí. Celkově lze říci, že tento kód HTML tvoří strukturu a obsah webové stránky, který je poté zobrazen v prohlížeči a umožňuje uživateli procházet a vyhledávat informace na stránce.

V další části kódu HTML se nachází element s třídou `"Kontakty"`, což představuje sekci stránky s odkazy na sociální sítě, konkrétně Instagram, YouTube a Discord. Tyto odkazy jsou zadány pomocí elementu `a` s atributem `href`, který umožňuje přeměřovat uživatele na danou stránku sociální sítě. Dále zde můžeme vidět elementy `img` s atributem `src`, které zobrazují ikony sociálních sítí a umožňují tak uživateli snadno najít odkazy na sociální sítě.

Následuje další sekce s elementem s třídou `"Blog"`, kde uživatel nalezne sekci stránky s informacemi o aplikaci MyDay a jejím týmu. Zde lze nalézt další obrázky (`img`) a odkazy (`a`) na další stránky nebo soubory. Kód HTML pokračuje zobrazením dalších sekcí stránky s různými prvky a elementy, jako jsou obrázky, tabulky, formuláře a další. V závěru kódu HTML se nachází element `/html`, což ukončuje samotný kód HTML a umožňuje prohlížeči zobrazit obsah stránky. V celku lze říci, že tento kód HTML tvoří strukturu a obsah webové stránky, který je poté zobrazen v prohlížeči a umožňuje uživateli procházet a vyhledávat informace na stránce.

CSS

kód CSS začíná sekcí `* { ... }`, což znamená, že se změny týkají všech prvků na stránce. V této sekci je nastavena vlastnost `box-sizing` na hodnotu `"border-box"`, což znamená, že velikost prvku se bude počítat včetně okrajů (`border`) a výplně (`padding`). Všechny prvky na stránce budou také mít nulový `margin`, tedy mezeru kolem sebe. Následuje sekce `body { ... }`, která se týká prvku `body`, tedy celého obsahu stránky. V této sekci je nastaven font písma na `"Roboto"` a barva písma na bílou. Je zde také použit efekt `linear-gradient`, který vytváří přechod mezi více barevnými odstíny na pozadí stránky. Efekt je animován pomocí `@keyframes` a třídy `"gradient"`, což umožňuje neustálé opakování animace. Dále jsou zde

sekce s různými třídami, jako například "nav-container", "nav", "header-container" atd., které slouží k definování stylů pro různé prvky na stránce. Tyto třídy jsou přiřazeny k náležitým prvkům v html.

```
JS main.js > ...
1  const searchIcon = document.querySelector(
2    ".search-icon-container .search-icon"
3  );
4  const searchInput = document.querySelector(".search-input-container input");
5  const closeIcon = document.querySelector(".search-icon-container .close-icon");
6  const mobileCloseIcon = document.querySelector(
7    ".mobile-menu-container .close-icon"
8  );
9  const mobileMenuContainer = document.querySelector(".mobile-menu-container");
10 const menuIcon = document.querySelector("nav .menu-icon");
11
12 searchIcon.addEventListener("click", () => {
13   searchInput.classList.add("active");
14   closeIcon.classList.add("active");
15   searchIcon.classList.remove("active");
16 });
17
18 closeIcon.addEventListener("click", () => {
19   searchInput.classList.remove("active");
20   closeIcon.classList.remove("active");
21   searchIcon.classList.add("active");
22 });
23
24 menuIcon.addEventListener("click", () => {
25   mobileMenuContainer.classList.add("active");
26 });
27
28 mobileCloseIcon.addEventListener("click", () => {
29   mobileMenuContainer.classList.remove("active");
30 });
```

Obrázek 19 Ukázka JavaScript kódu (zdroj: autor)



Obrázek 20 Aktuální snímek obrazovky titulní strany (zdroj: autor)

Aplikace na pomoc lidem s psychickými poruchami.

Na jaké poruchy se zaměřujeme?

Obrázek 21 Obrazovka po kliknutí na zjistit více (zdroj: autor)

Závěr

Navrhli jsme uživatelsky přívětivou android mobilní aplikaci. Její podobu jsme několikrát v průběhu jejího vývoje měnili do podoby takové, která je pro oko uživatele uklidňující a zároveň jednoduchá. Jsme si vědomi, že naši klienti ji budou užívat ve chvílích, kdy se nebudou cítit nejlépe. Naším cílem bylo, aby aplikace byla uživatelům k dispozici v případě lehkých až těžkých obtíží, jímž jsou deprese, panické ataky a různé formy úzkosti. Navrhli jsme systém čtyř postupů pomoci, jež byla původním cílem, v závislosti na závažnosti stavu uživatele. V průběhu vývoje jsme zjistili, že čtyři postupy nebudou dostatečné, a proto jsme zavedli pátý postup při extrémních případech. Tento postup je zpracován formou telefonického spojení s linkou první psychické pomoci. Věříme, že jsme cíle splnili dostatečně i přes různé komplikace ve vývoji.

Webový portál je navržen v co nejjednodušší podobě, aby i lidem, kteří nejsou technicky příliš zdatní, umožnil se lidem v problematice zorientovat, získat základní informace a umožnit potenciálnímu uživateli aplikaci stáhnout. Toto byl jeden z našich základních cílů, kterého se nám, věříme, podařilo dosáhnout.

Vytvořili jsme nástroj, jenž je možné v budoucnosti dále přizpůsobovat aktuálním potřebám, odborným stanoviskům a novým technologiím. V budoucnu bychom na projektu rádi pracovali a snažili se v něm promítat zkušenosti z jeho fungování, ať již naše či uživatelů.

Shrnutí

Psychické poruchy jsou v posledních letech stále častější a jsou spojené s tlakem, který působí na každého z nás v současné rychlé době. To vedlo k projektu, který má pomoci lidem trpícím psychickými poruchami a obohatit společnost. Cílem projektu bylo vytvořit aplikaci, která by byla přehledná a jednoduchá, a poskytovala ověřené způsoby pomoci v akutních situacích. Projekt také zahrnoval internetový portál, který poskytoval informace o vývoji projektu a možnost zpětné vazby od uživatelů. Projekt se zaměřil na tři konkrétní poruchy: panickou poruchu, depresi a úzkosti, a obsahoval osobní poznatky spojené s ověřenými informacemi od WHO a kontroly od psychologa a psychiatra.

Projekt pokračoval vyhledáváním informací o prostředcích potřebných k vytvoření odborné práce a vytvářením a testováním aplikace a internetového portálu. Výsledkem bylo vytvoření účinného nástroje pro pomoc lidem trpícím psychickými poruchami. Celkově projekt představuje užitečné řešení pro lidi trpící psychickými poruchami a přispívá k obohacení společnosti. Jeho cílem je poskytnout pomoc v akutních situacích a umožnit lidem s psychickými poruchami získat potřebné informace a podporu. Aplikace a internetový portál jsou navrženy tak, aby byly přehledné a jednoduché a byly vyvinuty s cílem pomoci lidem v kritických situacích. Jeho cílem je poskytnout kvalitní pomoc lidem trpícím psychickými poruchami a pomoci jim v boji s těmito nemocemi.

Použité zdroje informací

1. Depression. *Who.int* [online]. International: WHO, 2021 [cit. 2022-12-11]. Dostupné z: <https://www.who.int/news-room/fact-sheets/detail/depression>
2. Panic disorder. *Nhs.uk* [online]. UK: NHS, 2020 [cit. 2022-12-11]. Dostupné z: <https://www.nhs.uk/mental-health/conditions/panic-disorder/>
3. Anxiety disorders. *Paho.org* [online]. International: PAHO, 2021 [cit. 2022-12-11]. Dostupné z: <https://www.paho.org/en/noncommunicable-diseases-and-mentalhealth/noncommunicable-diseases-and-mental-health-data-portal-2>
4. DATTANI, Salon, Hannah RITCHIE a Max ROSER. Mental and substance use disorders are common globally. *Our World in Data* [online]. 2018, **2021**(1) [cit. 2023-01-02]. Dostupné z: <https://ourworldindata.org/mental-health>
5. Mental Illness. *National Institute of Mental Health* [online]. USA: NIMH, 2022 [cit. 2023-01-02]. Dostupné z: <https://www.nimh.nih.gov/health/statistics/mental-illness>
6. Substance Abuse and Mental Health Services Administration. *SAMHSA* [online]. USA: Samhsa [cit. 2023-01-02]. Dostupné z: <https://www.samhsa.gov/data/>
7. Mental Health By the Numbers. *NAMI* [online]. USA: Samhsa, 2022 [cit. 2023-01-02]. Dostupné z: <https://www.nami.org/mhstats>
8. Population, total - United States. *The World Bank* [online]. USA: The World Bank, 2021 [cit. 2023-01-02]. Dostupné z: <https://data.worldbank.org/indicator/SP.POP.TOTL?end=2021&locations=US&start=1960>
9. ČÁPKA, David. Lekce 1 - Úvod do programovacího jazyka C. *The World Bank* [online]. Praha, Česká republika: ITNetwork, 2019 [cit. 2023-01-02]. Dostupné z: <https://www.itnetwork.cz/cecko/zaklady/tutorial-uvod-do-programovacihojazyka-c>
10. Nepanikař - První pomoc při psychických potížích. Nepanikař - První pomoc při psychických potížích [online]. Dostupné z: <https://nepanikar.eu/>
11. VOS: Wellbeing & Intelligent Journal - Home. VOS: Wellbeing & Intelligent Journal - Home [online]. Dostupné z: <https://vos.health/>
12. Visual Studio Code – Wikipedie. [online]. Dostupné z: https://cs.wikipedia.org/wiki/Visual_Studio_Code
13. Adobe XD - Wikipedia. [online]. Dostupné z: https://en.wikipedia.org/wiki/Adobe_XD