

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Tvorba uživatelských rozhraní
Technická zpráva - Machala Roman (xmacha86)

1 Úvod

Na základě zpětné vazby pitch prezentace, jsme se jako tým dohodli implementovat 2 verze námi navržené aplikace. Jednotlivé verze jsou implementovány nezávisle na sobě.

Z důvodu vývoje 2 verzí aplikací je následně popsána aplikace a všechny příložené soubory s touto dokumentací autorským dílem Romana Machaly (xmacha86). Jelikož React Native, ve kterém je aplikace implementována silně spojuje logiku view a controlleru v rámci komponent, kdy oddělení těchto částí není tak striktní jako v tradičním MVC modelu, vede tato aplikace spíše k MVVM¹ modelu.

2 Základní myšlenka aplikace

Aplikace byla vyvíjena se záměrem poskytnout aplikaci především studentům, kteří jsou poměrně časově vytížení a mnohdy mají problémy s časovou organizací². Použití má být především jednoduché, rychlé a aplikace má být uživatelsky přívětivá, aby ji dokázal uživatel pochopit, i když ji vidí poprvé. Aplikace umožňuje vkládat pravidelné události do části Scheduler, viz sekce 5, např. přednášky, cvičení, hobby, týmové meetingy atd. Druhou částí aplikace je TODO - List, viz sekce 4, která umožňuje vkládat nadcházející úkoly např. písemky, zkoušky, deadlines projektů, Kromě vkládání jednotlivých položek do obou částí je možno tyto položky následně upravit, pokud například dojde k překlepu při zadávání například názvu události. Všechny položky je následně možno odstranit v případě, že byl úkol dokončen, nebo již daná položka pro uživatele nemá význam. Pro jednotlivé položky je možno vybrat jejich barvu, se kterou se následně budou zobrazovat. Úkoly zadávané do TODO - Listu mají ještě atribut prioritita, udávající prioritu úkolu. Dle této priority jsou následně řazeny všechny úkoly, s největší prioritou nahoře a nejnižší dole.

3 Hlavní tělo aplikace

Hlavní tělo aplikace je uloženo v souboru `Body.js` nacházející se ve složce `./mainComponents/`. Tato část implementuje tabulátorovou navigaci pomocí prostředí `NavigationContainer`, jejíž účelem je přepínání mezi jednotlivými částmi aplikace (Scheduleru a TODO - Listu). Implicitně nastavenou stránkou při zapnutí aplikace je Scheduler. Přepínání mezi jednotlivými částmi se pak provádí stisknutím příslušného tlačítka ve spodní části aplikace. Při kliknutí na jedno z tlačítek je předáno jméno (route) dané stránky, kterou chceme zobrazit, funkci `screenOptions` jako parametr, která následně vynutí překreslení obrazovky na odpovídající stránku. Součástí tohoto prostředí je nadále stylizace a chování jednotlivých tlačítek při jejich stisknutí.

4 TODO - List

Část TODO - List je implementována v souboru `ToDoList.js` nacházející se ve složce `./mainComponents/screens/`. Záznamy o jednotlivých úkolech jsou ukládány do pole, kde unikátním klíčem každé položky je index v poli. Každý úkol je dále implementován pomocí prostředí `TouchableOpacity`.

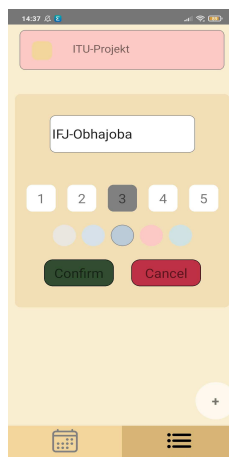
4.1 Vkládání úkolů

Po stisknutí tlačítka v pravém spodním rohu je pomocí metody `onPress` zavolána funkce `handleOpenModal` vynucující zobrazení modálního okna pro vkládání jednotlivých položek.

Uživatel pak může vložit název svého úkolu, vybrat si danou prioritu a barvu. Všechna tato tlačítka při stisknutí pomocí metody `onPress` zavolají příslušnou funkci, která zpracuje vybrané parametry daného úkolu. Po stisknutí tlačítka Confirm je metodou `onPress` zavolána funkce `handleAddTask`, která přidá novou položku do pole položek. Součástí funkce je zároveň seřazení pole dle priority položek pomocí metody `sort` nad polem položek a je vynuceno vykreslení aktuálních položek, které je provedeno pomocí metody `map`. Stisknutí tlačítka Cancel se vynutí zavření modálního okna a všechny provedené změny jsou zahozeny.

¹Model-View-ViewModel

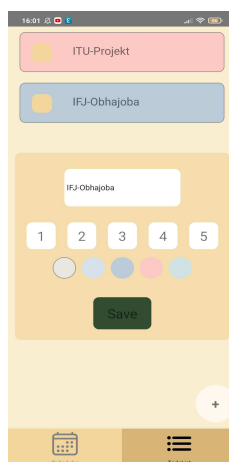
²Vycházím z průzkumu uživatelských potřeb



Obrázek 1: Modální okno pro přidání úkolu

4.2 Editace úkolů

Všechny vložené úkoly jsou implementovány pomocí prostředí `TouchableOpacity`, tudíž kliknutím na ně, pomocí metody `onPress` je zavolána funkce `openEditModal` vynucující otevření modálního okna pro editaci. Editace následně funguje podobným způsobem jako přidávání úkolů. Při stisknutí daného tlačítka je zavolána odpovídající funkce zpracovávající daný parametr, který chceme změnit. Po stisknutí tlačítka `Confirm` je zavolána funkce `handleEditTask`, která jako parametr dostane unikátní klíč dané položky (její index) a následně tuto položku vyhledá a provede aktualizaci novými atributy. Součástí funkce `handleEditTask` je opět seřazení daného pole dle priority a jeho následné vykreslení, obdobně jako u vkládání.



Obrázek 2: Modální okno pro editaci úkolu

4.3 Mazání úkolů

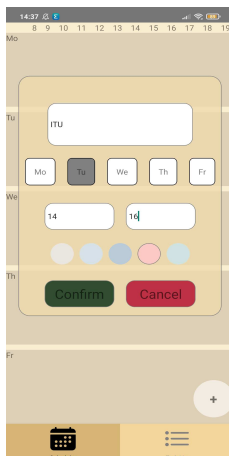
Smazání úkolu je provedeno kliknutím na oranžový čtvereček nacházející se na levé straně jednotlivých úkolů, viz obrázek 5. Pomocí metody `onPress` je zavolána funkce `completeTask`, která jako parametr dostane index daného úkolu, ten následně vyhledá a odstraní. Dojde opět k vyobrazení ostatních úkolů a k jejich přeřazení na základě priority.

5 Scheduler

Část Scheduler je implementována v souboru `Scheduler.js` nacházející se ve složce `./mainComponents/screens/`. Jednotlivé položky jsou uchovávány v poli 5-ti polí, kde každé z 5-ti polí představuje 1 den od Po do Pá. v každém tomto poli jsou následně vedeny záznamy o jednotlivých událostech pro tento den. Unikátním klíčem každé položky je pak kombinace dne a indexu v poli dnů.

5.1 Vkládání úkolů

Vkládání úkolů začíná stisknutím tlačítka v pravém dolním rohu, kdy pomocí metody `onPress` zavolána funkce `handleOpenModal`, která vynucuje otevření modálního okna pro vkládání nových položek. Uživatel si zde může



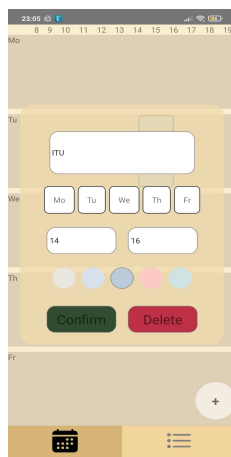
Obrázek 3: Modální okno pro přidání položky

zvolit název události, den ke kterému má být událost připojena, začátek a konec události a vhodnou barvu. Kliknutím na jednotlivá tlačítka dochází k zavolání příslušné funkce zpracovávající daný parametr nově vkládaného úkolu. Po stisknutí tlačítka `Confirm` je zavolána prostřednictvím metody `onPress` funkce `handleAddEvent`, která přidá položku ke správnému dni. Součástí této funkce jsou ještě podpůrné funkce, jako je například `calculatePosition`, která na základě požadované začínací a končící doby události vrátí horizontální posunutí v daném dni. Vertikální umístění je dáno dnem, ke kterému událost patří.

5.2 Editace úkolů

Editace úkolů funguje podobně jako vkládání. Jednotlivé položky jsou implementovány pomocí prostředí `TouchableOpacity`, kdy po jejich stisknutí je zavolána funkce `openEditModal`, která vynucuje otevření modálního okna pro editaci. Uživatel zde může změnit jednotlivé atributy vybrané položky pomocí jednotlivých tlačítek nebo textových vstupů, po jejichž kliknutí nebo změně je opět zavolána příslušná funkce zpracovávající nové parametry. Po tliknutí na tlačítko `Confirm` je pomocí metody `onPress` zavolána funkce `handleEditEvent`, která aktualizuje vybranou položku³. Součástí této funkce je zároveň vynucené překreslení zobrazovaných položek a v případě nutnosti přepočet horizontálního posunutí v daném dni, nebo přesun této položky do dne jiného.

³Daná položka je vyhledána pomocí kombinace dne a indexu v poli dnů



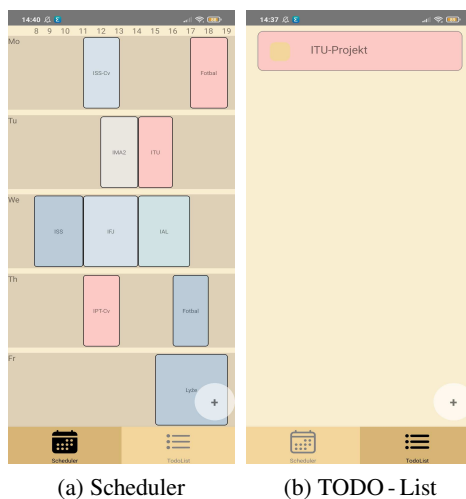
Obrázek 4: Modální okno pro editaci položky

5.3 Mazání úkolů

Mazání úkolu probíhá pomocí modálního okna pro editaci, viz sekce 5.2. Pomocí stisknutí tlačítka Delete je pomocí metody `onPress` zavolána funkce `removeActivity`, která následnou položku vyhledá a odstraní. Součástí je i znovuvyobrazení aktuálních položek ve Scheduleru.

6 Výsledná aplikace

Na následujících obrázcích je možno vidět výsledný vzhled aplikace, jak by mohl vypadat při několika vložených položkách.



Obrázek 5: Ukázka aplikace