Fakulta riadenia a informatiky
Informatika

Vývoj aplikácií pre mobilné zariadenia

Dokumentácia

Benjamín Koša, 5ZYI23

Obsah

Kalkulačka prevodových pomerov na bicykli – “Bike Gear Ratio”

Jedná sa o jednoduchú aplikáciu na ktorej si je možné navoliť veľkosť kolesa, rozsah a počet zubov na kazete a na stredovom prevodníku. Podľa navoleného vstupu sa urobí výpočet prevodových pomerov. Výsledok sa uloží do matice a zobrazí užívateľovi ako tabuľka. Výslednú tabuľku si môže užívateľ uložiť a pozrieť neskôr.

Aplikáciu sme sa rozhodli vyvíjať v React Native pre obe platformy Anodrid a iOS.

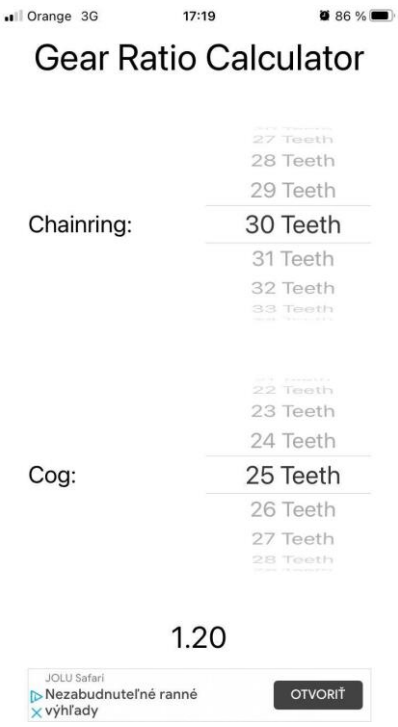
Výpočet prevodových pomerov

Na meranie sme sa rozhodli použiť metódu “Metres of development”. Výpočet sa robí podľa jednoduché vzorca:

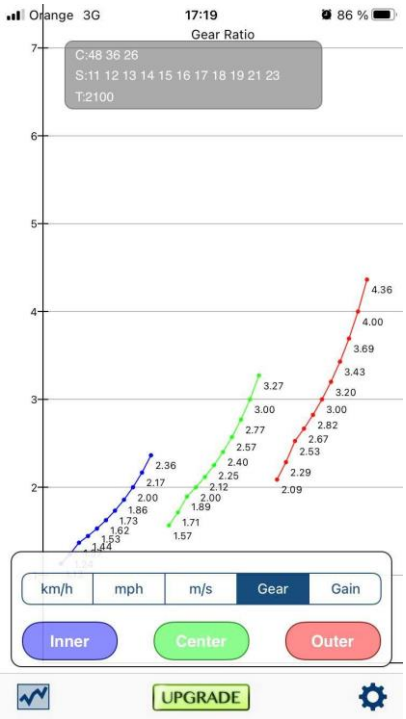
$$x = \text{obvod kolesa v metroch} * \frac{\text{počet zubov na stredovom pevodníku}}{\text{počet zubov na kazete}}$$

Spracovanie prehľadu dostupných aplikácií podobného zamerania

Vybrali sme si štyri aplikácie ktoré sa dali stiahnuť z apple app store zadarmo.

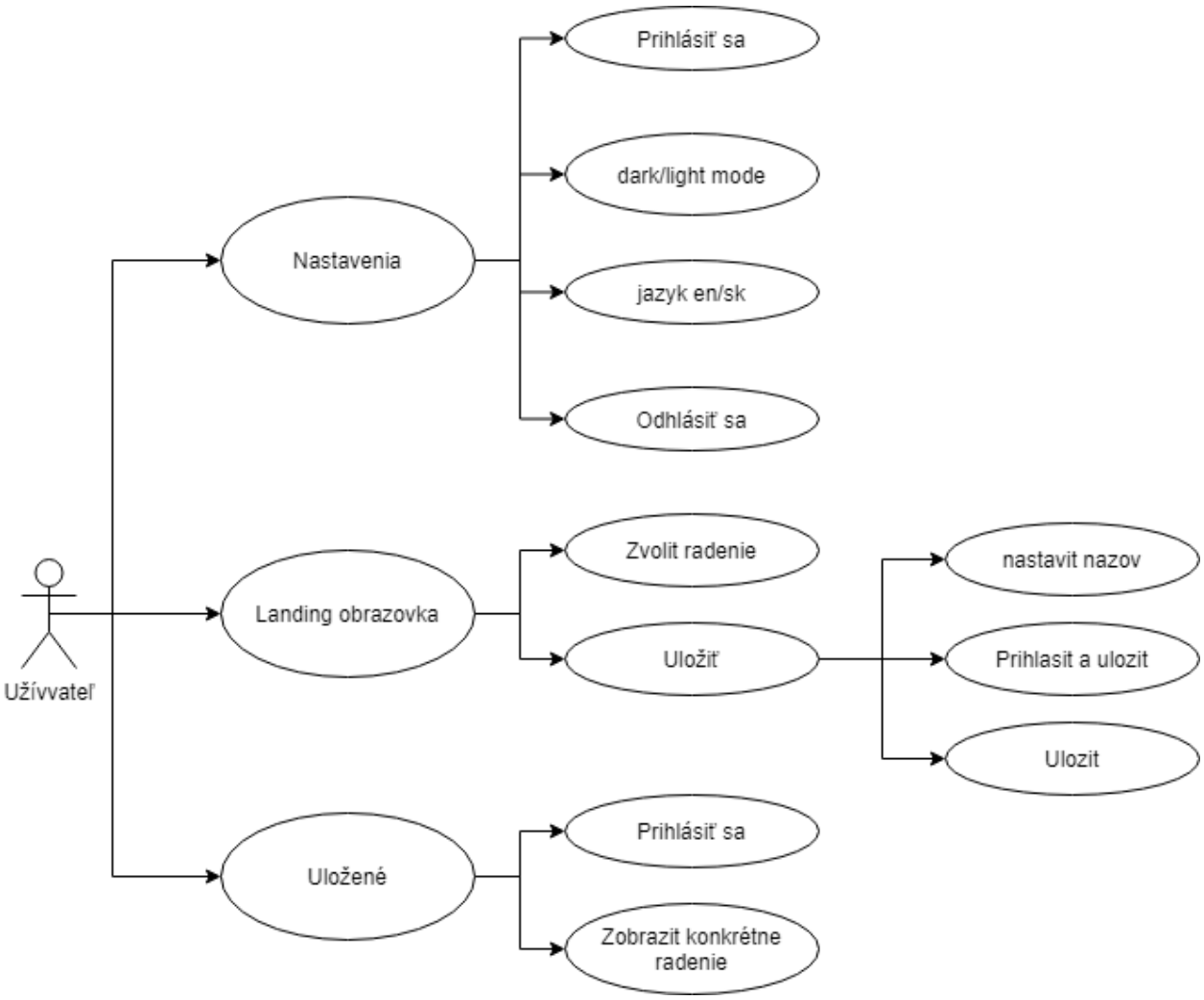


V týchto troch aplikáciách je možne si navoliť len konkrétny jeden prevod.



V tejto aplikácii si užívateľ môže navoliť kompletné radenie a pozrieť si pekne vykreslené pomery v grafe. Žiadna z aplikácii neponúka možnosť uložiť si navolené radenie.

Analýza navrhovanej aplikácie - use case diagram

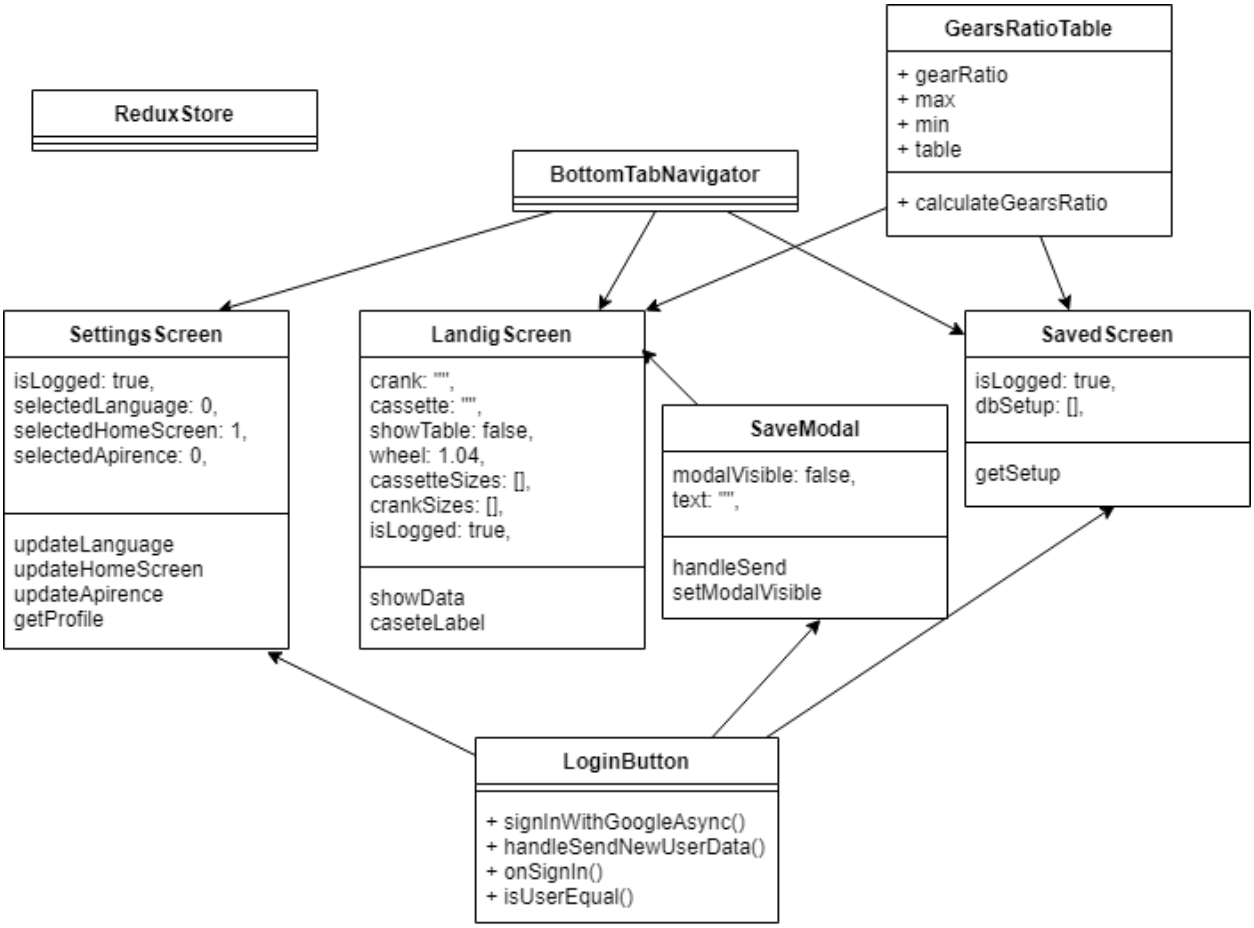


Pri prihlasovaní pomocou google, z pohľadu užívateľa, nerozlišujeme registráciu a prihlásenie.

Návrh architektúry aplikácia

Aplikácia sa skladá z troch hlavných obrazoviek nastavenia, domáca obrazovka(landing) a uložené. Prepínanie medzi nimi zabezpečuje [TabNavigátor] ktorý je umiestnený na spodnej časti obrazovky.

Diagram tried



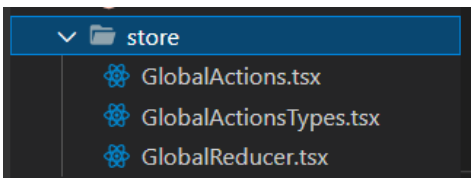
ReduxStore

Je pripojený ku každej treide ale kvôli prehľadnosti sme ho nechali takto.

Slúži na uloženie dát ktoré sa používajú v celej aplikácii, u nás sú to iba nastavenia. Vďaka redux sa zmena nastavení prejaví hneď bez nutnosti reštartovania aplikácie.

Je to takzvaný kontajner na state aplikácie.

Store sa skladá z troch súborov Actions, ActionsTypes a Reducer



V Actions su definované všetky metódy nad storom, v jednoduchosti slúžia na pridávanie dát do storu, niečo ako setter. Na nasledujúcom obrázku je príklad akcie ktorá zmení jazyk v aplikácii.

```
export const setLanguage = (language: any) => ({
  type: SET_LANGUAGE,
  payload: language,
});
```

V Reducer sa vykoná akcia a prepíšu sa dáta v store.

```
const globalReducer = (state = INITIAL_STATE, action: any) => {
  switch (action.type) {
    case ADD_FRIEND: ···
    case ADD_USER: ···
    case SET_APIRENCE: ···
    case SET_HOMESCREEN: ···
    case SET_LANGUAGE:
      const lang: { [key: string]: string } = { "0": "sk", "1": "en" };
      let newLanguageState = { ...state };

      newLanguageState.selectedLanguage = action.payload;
      newLanguageState.appLang = lang[action.payload + ""];

      return newLanguageState;
    default:
      return state;
  }
};
```

V ActionTypes sú definovane všetky typy akcii, kvôli tomu aby sme sa vyhli magickým hodnotám ktoré sa využívajú v súbore Recuer a Actions.

```
1  export const ADD_FRIEND = 'ADD_FRIEND';
2  export const ADD_USER = 'ADD_USER';
3
4  export const SET_LANGUAGE = 'SET_LANGUAGE';
5  export const SET_HOMESCREEN = 'SET_HOMESCREEN';
6  export const SET_APIRENCE = 'SET_APIRENCE';
```

Obrazovka nastavenia SettingsScreen

Užívateľ sa môže prihlásiť, odhlásiť, zmeniť farebný mód na svetlý alebo tmavý, nastaviť jazyk.

Na uloženie nastavení, aby sa hneď prejavili v celej aplikácii využívame store React Redux.

```
/**
 * ulozi jazyk do asyncStorage a do store
 * @param selectedLanguage
 */
updateLanguage(selectedLanguage: number) {
  this.setState({ selectedLanguage });
  AsyncStorage.setItem("selectedLanguage", selectedLanguage + "");
  this.props.setLanguage(selectedLanguage);
}
```

Ukážka ukladania nastavenia jazyka, nastavenie sa uloží do state daného komponentu, do async storage aby ostalo uložené aj po vypnutí aplikácie a volaním metódy props.setLanguage() do redux store.

Obrazovka domáca obrazovka LandingScreen a SaveModal

Užívateľ si môže navoliť radenie, používame react-native-picker-select a sprístupní sa mu button uložiť. Po kliknutí na uložiť sa vyroluje modal v ktorom si navolí názov radenia možnosť uložiť alebo prihlásiť a uložiť.

Obrazovka uložené SaveScreen

Ak užívateľ nieje prihlásený zobrazí sa hláška a tlačidlo na prihlásenie.

V opačnom prípade sa vytiahne z databázy zoznam všetkých uložených radení daného užívateľa a zobrazí v zozname. Po kliknutí na položku sa zobrazí tabuľka prevodových pomerov pre dané radenie.

```
/**
 * subscribe na uzivatelove ulozene prevody
 * v databaze
 */
getSetup = () => {
  const token = firebase.auth().currentUser?.uid;
  if (token) {
    this.firestoreUnsubscribe = firebase
      .firestore()
      .collection("setup_" + token)
      .onSnapshot((querySnapshot) => {
        this.setState({
          dbSetup: querySnapshot.docs.map((doc) => ({
            ...doc.data(),
            id: doc.id,
            icon: "gears",
            showTable: false,
          })),
        });
      });
  }
};
```

Ukážka metódy ktorá vytiahne z databázy uložené prevody prihláseného užívateľa. Každý užívateľ ma originálne id podľa ktorého sú v databáze uložene jeho údaje. Dáta chodia vo formáte json.

Vlastné komponenty GearsRatioTable a LoginButton

Vytvorili sme komponent pre tlačidlo prihlásenie pretože sa využíva na viac miestach aplikácie a komponent tabuľky prevodových pomerov z toho istého dôvodu.

V react native sa všetko označuje ako komponent, v kotline by to bol pravdepodobne fragment.

Prihlasovanie a registrácia

Prihlasovanie a aj databázu zabezpečuje firebase, konkrétne využívame nosql databázu firestore a firebase authentication pre prihlasovanie. Aplikáciu je možné používať aj bez internetového pripojenia ale nieje možné sa prihlásiť a uložiť.

Spustenie aplikácie

Expo

Najjednoduchší spôsob pre android a aj iOS je stiahnuť si aplikáciu Expo, naskenovať priložený qr kód a za pár sekúnd sa vám načíta testovacia plne funkčná verzia aplikácie.

Inštalácia apk

Na zariadeniach typu android je možné si klasickým spôsobom nainštalovať .apk súbor.

Link na stiahnutie:

https://expo.io/artifacts/0ce0914b-9ead-4bc9-9980-3d6406424568

https://studuniza-my.sharepoint.com/:u:/g/personal/kosa11_stud0_uniza_sk/EZH05JOw7dtMutZOPxvB71sBW68LGpSviyrGJ94XcKLZ5Q?e=wu1cUN

Kompilácia zdrojového kódu

Po naklonovaní projektu stačí v základom priečinku spustiť príkazy:

http://github.com/benkosa/BikeGears

- npm install
- expo install
- expo start

IDE odporúčame Visual Code

Záver

Sme radi že sme sa rozhodol naučiť práve framework react native. Veľká výhoda je že vyvíjame naraz na obidve najpoužívanejšie platformy. Počas vývoja sme sa nestretli so žiadnymi väčšími komplikáciami alebo chybami. Páči sa nám ako sú v reacte od seba logicky oddelené jednotlivé komponenty a akým spôsobom medzi sebou komunikujú. Dobre sme spravil že sme sa rozhodol pre silne typovo kontrolovanú verziu JavaScriptu TypeScript ktorý ukáže veľa chýb ešte pred skompilovaním.

Vďaka firebase sa dá rýchlo vytvoriť funkčný backend, databáza a aj prihlasovanie. Pracuje sa s ním jednoducho a hlavne pri menších projektoch sa oplatí použiť takúto technológiu pretože si nemusíme robiť od začiatku všetko sami