

## Popis aplikácie

Habsheet má pomôcť jednotlivcom a skupinám spravovať si svoje aktivity, sledovať si ako sa týmto aktivitám venujú, ale aj porovnávať sa s ďalšími ľuďmi s rovnakými aktivitami a upozorňovať ich ak na dané aktivity zabúdajú. Cieľovou skupinou tejto aplikácie sú teda všetci, ktorí majú záujem venovať sa nejakým aktivitám a majú problém nájsť si pre ne čas, odhodlanie alebo sú len zabudliví.

Každý užívateľ bude mať možnosť vytvoriť si súkromné aktivity, ktoré bude vidieť iba on, alebo vytvoriť (poprípade stať sa členom) skupiny, kde môže pozvať viac ľudí a vytvárať aktivity pre viacerých účastníkov. V každej skupine bude jeden vlastník, ktorý bude mať možnosť pozývať ľudí, vytvárať a upravovať aktivity a ďalej členovia, ktorí budú mať možnosť tieto aktivity plniť.

Aktivita bude mať nejaký interval, v ktorom sa má opakovať (denné, týždenné, mesačné), požadovaný počet splnení za tento interval a stručný popis opisujúci čo je cieľom danej aktivity. (príklad: Napi sa vody, opakuje sa denne, požadovaný počet splnení 4x za daný interval teda 4x za deň).

Vždy keď užívateľ splní cieľ aktivity, poznačí si tento úspech a aplikácia mu oznámi, či je už hotový, poprípade koľko krát ma za daný interval aktivitu ešte splniť (príklad: užívateľ sa napije, poznačí si to a aplikácia ho pochváli a upozorní, že dnes by sa mal napiť ešte aspoň 3x).

V momente, keď užívateľ splní všetky opakovania, ostatní členovia skupiny, ktorí ešte nemajú cieľ splnený dostanú notifikáciu, že iný užívateľ úspešne skončil a povzbudí ich, aby sa snažili aj oni. Notifikácia sa môže objaviť aj v priebehu dňa, ak sa užívateľ v daný deň ničomu nevenoval, aby mu pripomenula jeho ciele.

## Právomoci

- Neprihlásený
- Užívateľ
- Vlastník skupiny
- Habit manager

### **Neprihlásený**

Bude mať možnosť prečítať si o hlavnom Gro aplikácie, registovať sa alebo sa prihlásiť

### **Užívateľ**

Bežný člen nejakej skupiny, bude mať právo vidieť aktivity skupiny a plniť jej ciele.

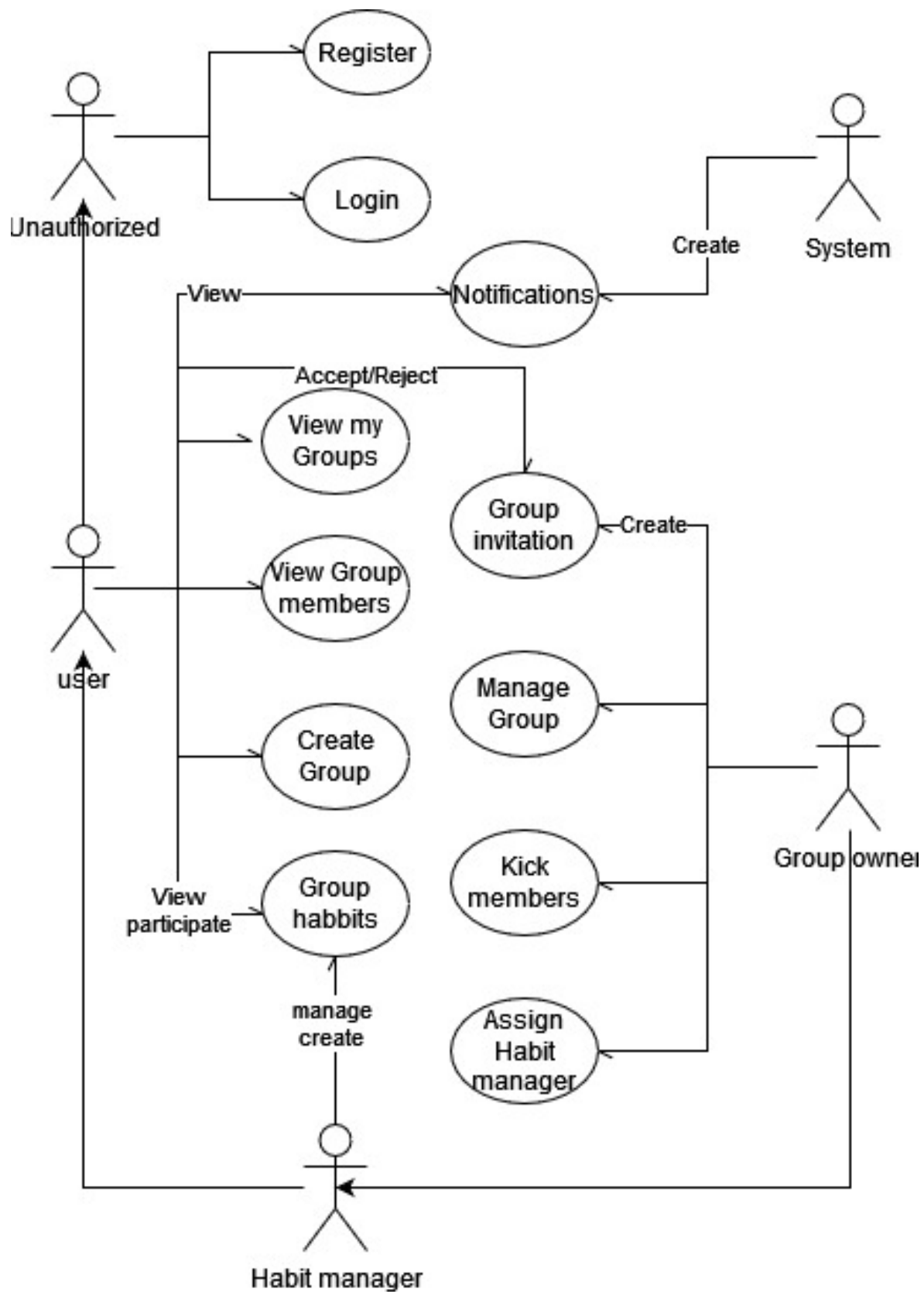
### **Habit Manager**

Bude mať právo vytvárať a managovať nové aktivity pre skupinu, ktorú nevlastní, ale v ktorej bol menovaný habit managerom.

### **Vlastník skupiny**

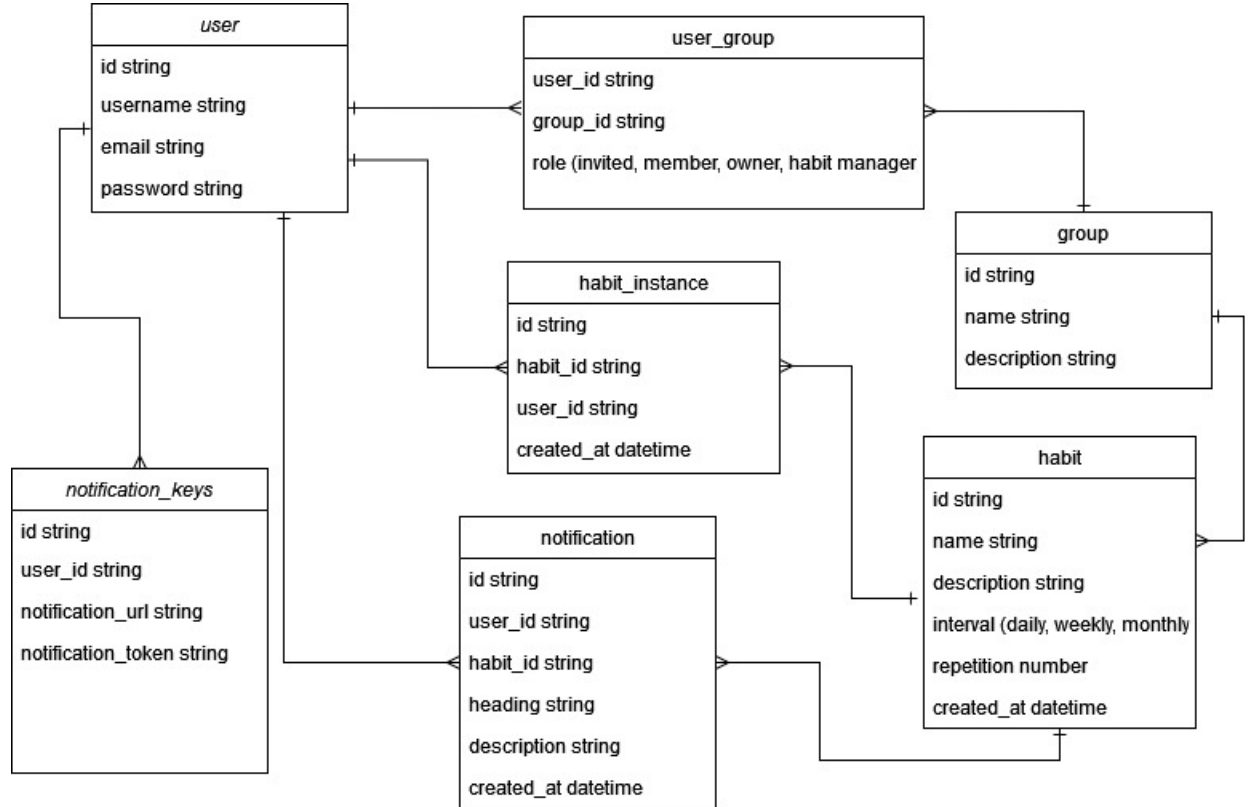
Bude mať právo pozývať nových členov do skupiny, vytvárať a managovať nové aktivity.

## Use case diagram



## Databáza

Ako databázu som sa rozhodol použiť noRDS databázu AWS DynamoDB. Rozhodol som sa o to hlavne kvôli jednoduchosti implementácie a možnosti používať databázu zadarmo v AWS (do istého limitu dát a potom typu platieb per request). Kvôli komfortu si ale zdefinujem štruktúru, ktorú plánujem dodržať.



## Popis tabuliek

**user** - základné informácie o každom užívateľovi

- **id** – užívateľa
- **username** – meno pod ktorým užívateľa uvidia ostatní
- **email** – unikátny email pod ktorým sa užívateľ prihlási
- **password** – zakryptované heslo užívateľa

**notification\_keys** – kľúče potrebné pre google push-up notifikácie (užívateľ ich môže mať viac, pretože môže mať viac zariadení napr. mobil, desktop)

- **id** – setu kľúčov pre notifikácie
- **user\_id** – id užívateľa ku ktorému sa kľúče vzťahujú
- **notification\_url** – url potrebná pre google push notifikácie
- **notification\_token** – token potrebný pre google push notifikácie

**user\_group** – prepojenie užívateľa a skupiny tabuľky taktiež obsahuje rolu, ktorú užívateľ plní v danej skupine

- **id** – prepojenia užívateľ/skupina
- **user\_id** – id užívateľa
- **group\_id** – id tabuľky
- **role** – enum obsahujúci aktuálne info o užívateľovi pre danú skupinu (pozvaný, užívateľ, vlastník, habit manager)

**group** – základné informácie o skupine

- **id** – skupiny
- **name** – meno skupiny
- **description** – stručný popis skupiny

**habit** – základné informácie o aktivite

- **id** – aktivita
- **name** – meno/hlavička aktivita
- **description** – stručný popis cieľa aktivita
- **interval** – v akom intervale sa má aktivita opakovať (denne, týždenne, mesačne)
- **repetition** – koľko krát treba splniť cieľ v danom intervale
- **created\_at** – kedy bola aktivita vytvorená (aby sme aktivitu nezobrazovali historicky kedy ešte neexistovala)

**habit\_instance** – po tom ako užívateľ splní cieľ aktivity vytvorí sa tu riadok, aby sa zaznamenal čas splnenia aktivity daným užívateľom

- **id** – aktivita
- **habit\_id** – id habitu na ktorý sa riadok viaže
- **user\_id** – id užívateľa na ktorého sa riadok viaže
- **created\_at** – kedy bol cieľ splnený

**notification** – základné informácie o notifikácii a informácia na akú aktivitu je cielená a pre akého užívateľa je určená (aby keď notifikáciu zmaže jeden užívateľ nech sa nezmaže aj ostatným)

- **id** – notifikácie
- **habit\_id** – id habitu na ktorý sa riadok viaže
- **user\_id** – id užívateľa na ktorého sa riadok viaže
- **heading** – nadpis/stručný popis notifikácie
- **description** – popis/obsah notifikácie
- **created\_at** – kedy bol cieľ splnený

## Technické požiadavky

Aplikácia bude používať react FE a aws serverless framework na strane BE (aws node js lambdy). Tie budú medzi sebou komunikovať pomocou REST API a aws API GATEWAY. Rozhodol som sa použiť typescript na strane FE aj BE pre jednoduchší vývoj a porozumenie kódu. Databázu budem používať dynamoDB. Pre používanie notifikácií plánujem používať google push notifikácie.

## Časový harmonogram

### 5. týždeň

- Dokončenie základnej kostry aplikácie (databáza + konfigurácia react/serverless)
- Dokončenie autorizácie a registrácie

### 6. týždeň

- Potvrdzovanie emailu
- Vytvorenie základnej funkcionality vytvárania skupín
- Vytvorenie základnej funkcionality vytvárania aktivít

### 7. týždeň

- Dokončenie skupín a aktivít
- Funkcionalita plnenia cieľov
- Pozývanie ľudí do skupín a potvrdzovanie pozvánok

### 8. týždeň

- Notifikácie (zatiaľ len vnútri aplikácie)
- Zobrazovanie historických dát
- Oprava detailov vzhľadu

### 9. týždeň

- Oprava detailov a vzhľadu
- Testovanie a príprava na betu aplikácie

### 10.-12. týždeň

- Práca na google push up notifikáciách
- Zobrazovanie historických dát v kalendáriu