

Modálne okná,  
trojvrstvová aplikácia

UINF/PAZ<sub>1c</sub>  
6. prednáška



kde sme?

dvojvrstvová aplikácia

SQL perzistencia

# Dvojvrstvová / two-tier aplikácia

## Prezentačná vrstva + aplikačná logika

- používateľské rozhranie
- zobrazenie a získavanie informácií od používateľa priamo prepojená s biznis logikou

## Databázová vrstva

- úložisko dát
  - typicky relačná databáza/súborový systém
- dáta bývajú často na samostatnom stroji

# Ako rozpoznať dvojvrstvovú aplikáciu?

- dáta sú uložené na **vzdialenom serveri**, všetko ostatné je v rámci aplikácie
  - „windowsovské účtovníctvo“
- **indikátor**: žonglovanie s dátami v obsluhu GUI
  - v metódach formulára sa zoskupujú/prepočítavajú dáta, ktoré prídu z DAO
- **indikátor**: vývoj alternatívneho UI/UX vyvoláva mdloby

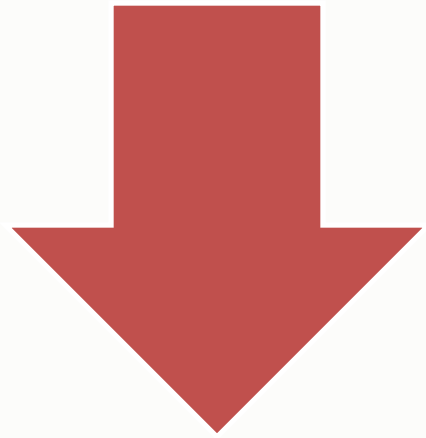
# Jednovrstvová aplikácia

- **single-tier** application
- „všetko na hromade“
- klasické skripty / úlohy z C
- aplikácia načíta zo súboru, požuje dáta, vypluje ich

Užitočné len v prípade jednopoužívateľských  
jednouúčelových aplikácií

biznis logika

Na koho zakričať „mám otvoriť  
používateľovi pri tejto čítačke“?

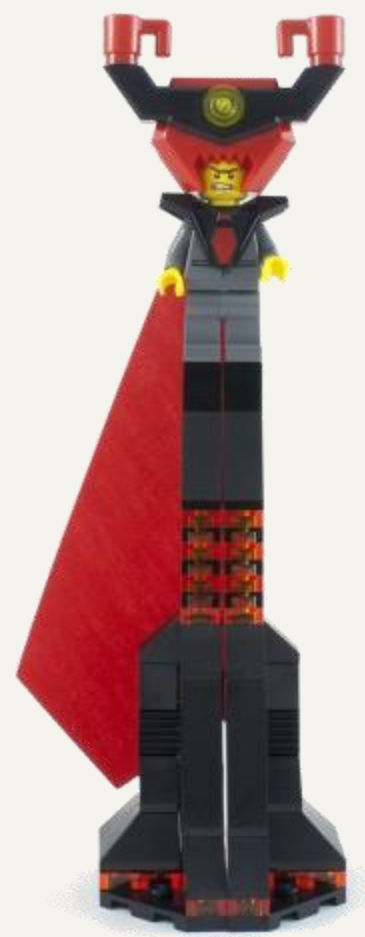
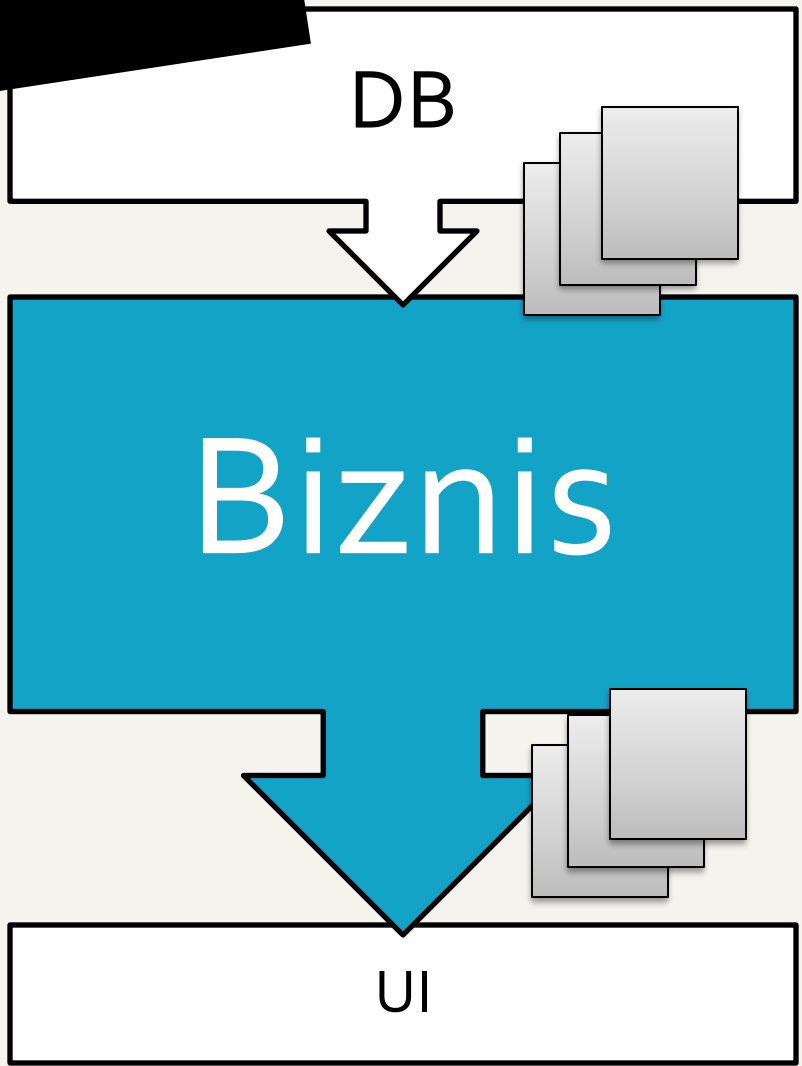


DAO?

Kontrolér?



mixovanie dát =  
biznis logika





# komplexné algoritmy?

- autorizácia na dverách
- nájsť, ktorí ľudia majú najviac práv
- ....

## Kód do obslužnej metódy vo formulári?

- čo ak sa UI zmení / rozšíri / vymení?
- čo ak chceme viacero rozličných UI?

Často používaný prístup, ale vedie k extrémne neprehľadnému kódu.



## Kód do metódy DAO?

- čo ak chceme viacero rozličných úložísk?
- čo ak chceme viacero variantov algoritmu?

Vhodný prístup pre jednoduché dvojvrstvové aplikácie, s výhradami.

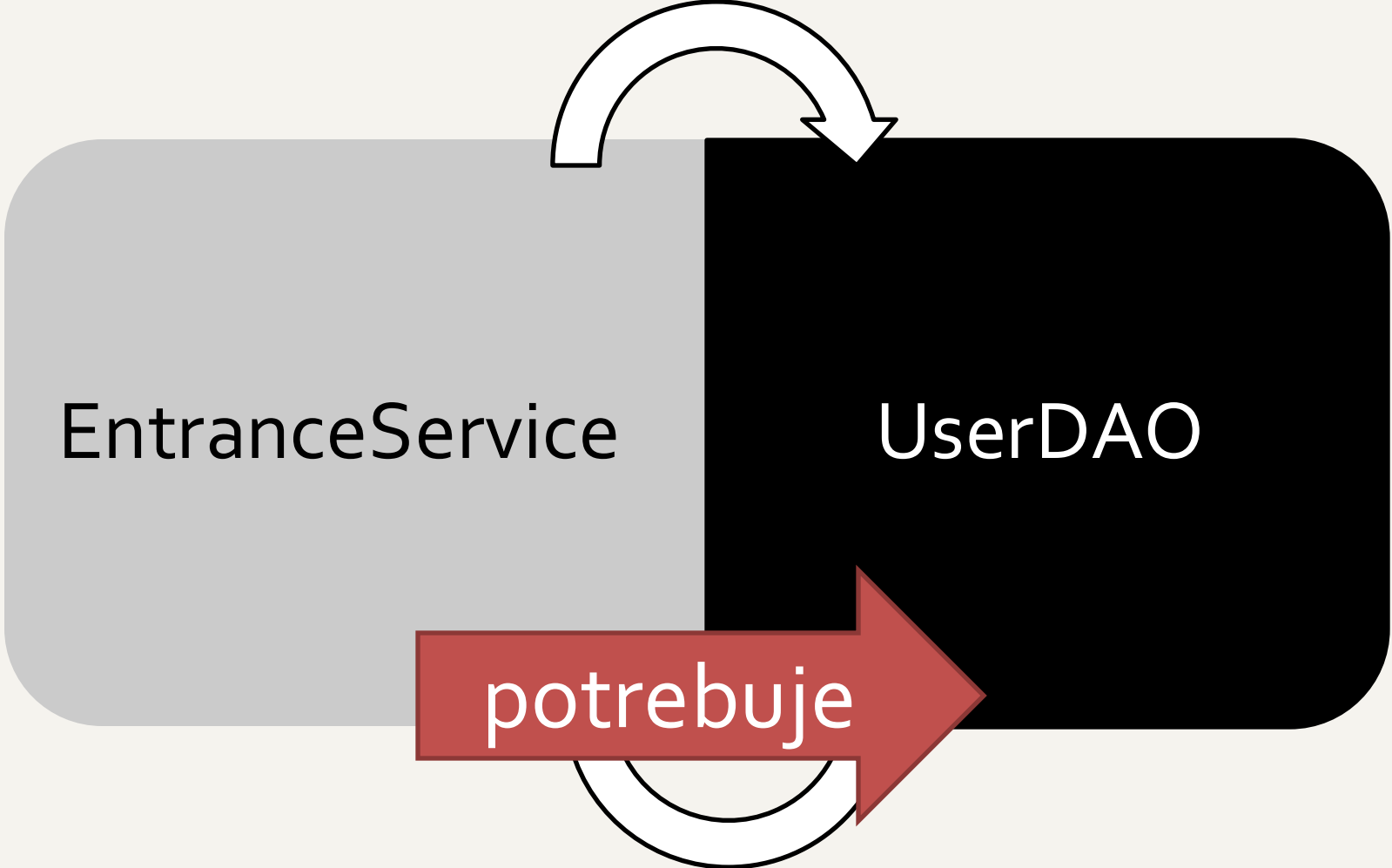


trieda biznis logiky

```
public interface EntranceService {  
    boolean validate(String chipId,  
        CardReader cr );  
}
```

```
public class DefaultEntranceService
    implements EntranceService {
...
    public boolean validate(String chipId,
        CardReader cr) {
        User user =...;
        if (user.isActive() &&
            user.hasAccess(CardReader cr)) {
            return true;
        }
        return false;
    }
}
```

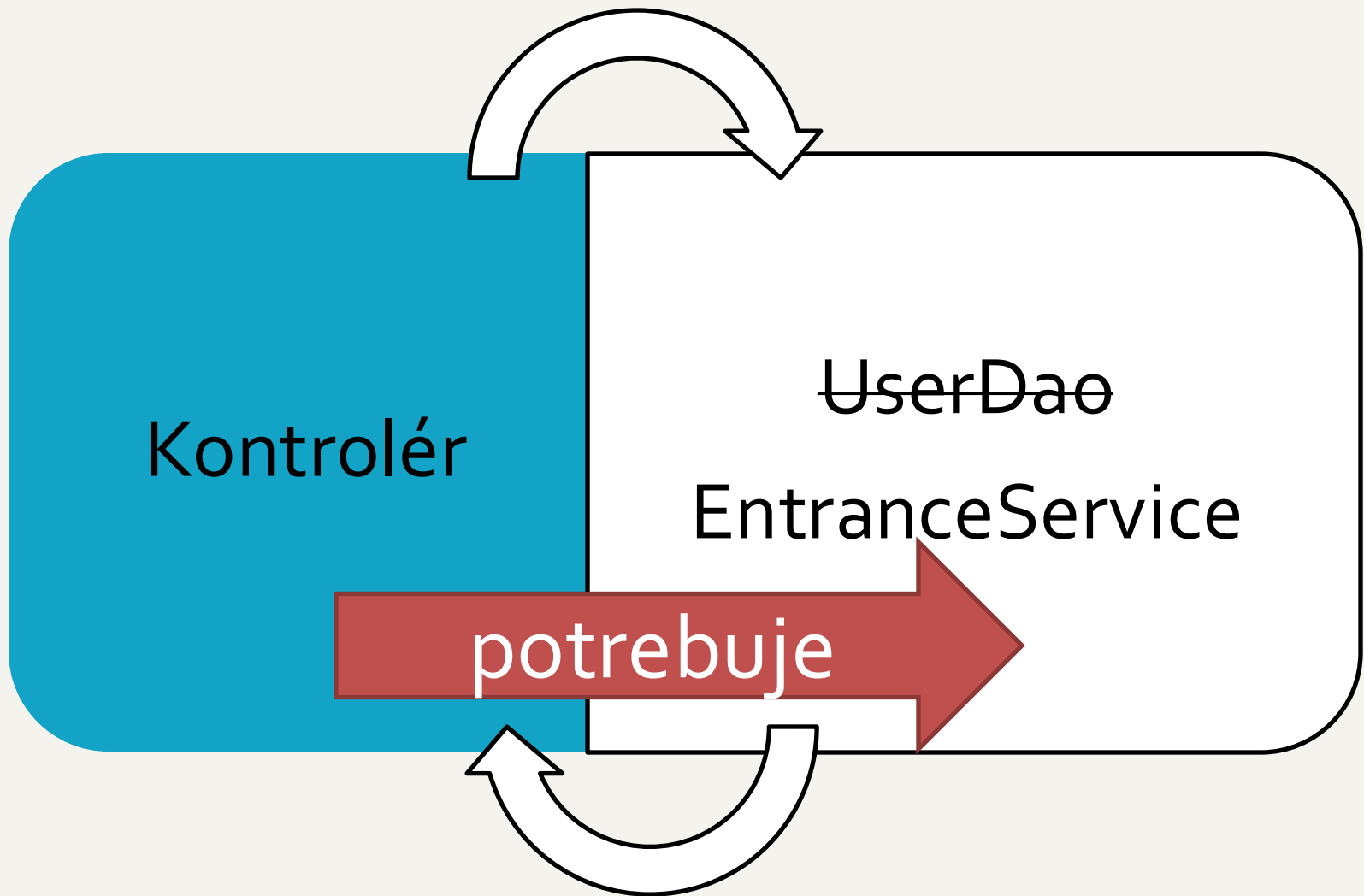
dáta pre službu?



```
public class DefaultEntranceService  
    implements EntranceService {
```

```
    private UserDao userDao  
        = UserDaoFactory.INSTANCE.getUserDao();
```

```
    public boolean validate(String chipId,  
        CardReader cr) {  
        User user = userDao.getByChipId(chipId);  
        /* overíme práva */  
    }  
}
```





Kontrolér

potrebuje

EntranceService

potrebuje

UserDao

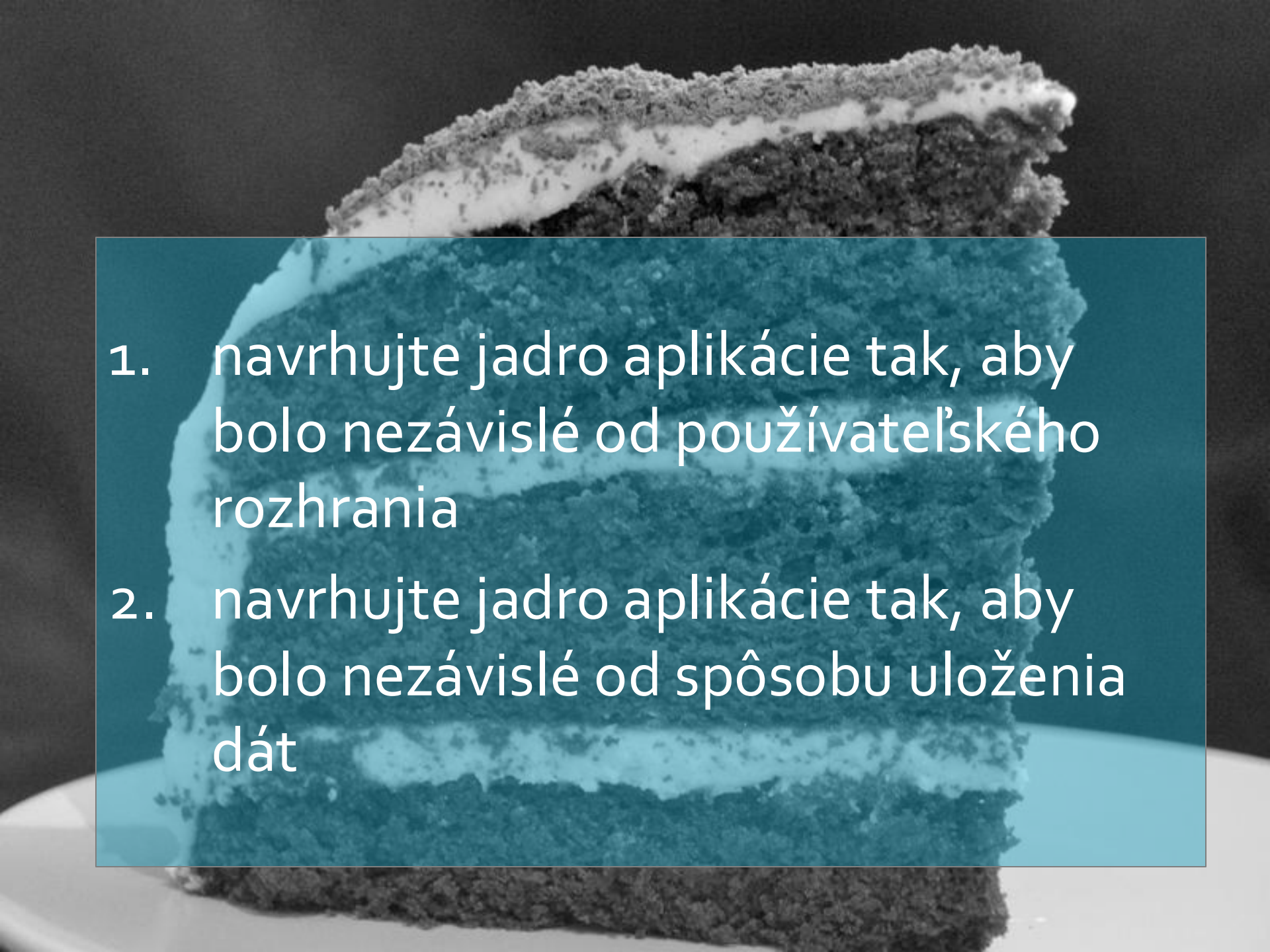
3tier

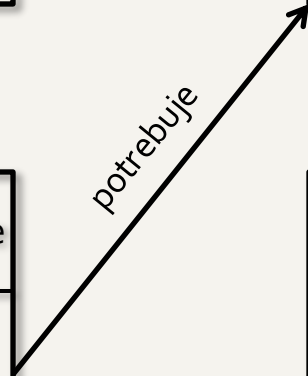
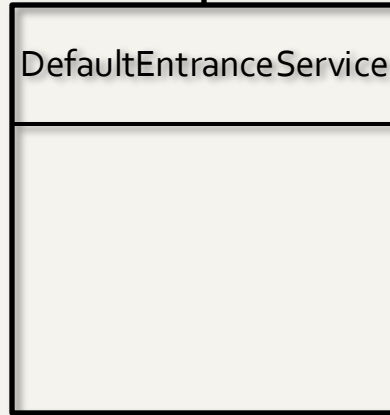
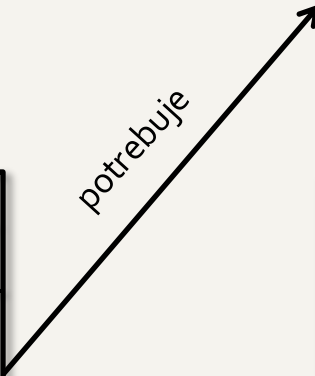
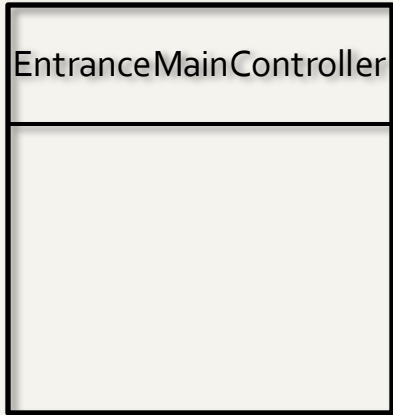
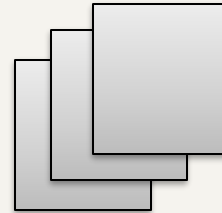
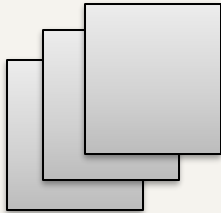
prezentácia  
UI/UX

biznis

príkazy,  
rozhodovanie,  
spracovanie

perzistencia  
uloženie dát

- 
1. navrhujte jadro aplikácie tak, aby bolo nezávislé od používateľského rozhrania
  2. navrhujte jadro aplikácie tak, aby bolo nezávislé od spôsobu uloženia dát



Otázky?