# PD1

## UZDEVUMS (28.oktobris)

1. Vispārīga lapas struktūra
2. Lietotājs var reģistrēties vietnē ar savu facebook kontu – informācija tiek glabāta MySQL datu bāzē
3. Jāizveido PHPUnit testi izstrādātajai funkctionalitātei
4. Nodrošināt žurnalēšanu, kļūdu paziņojumi tiek glabāti failā
5. Risinājums tiek iesūtīts izmantojot versiju kontroles sistēmu
6. Neliels paveiktā darba apraksts

## īss sistēmas apraksts

Šobrīd visus nepieciešamos failus iespējams lejuplādēt no mana git repozitorija:

<https://github.com/knospina/PD/> - repozitorijs ir publisks, tādēļ nav nepieciešams norādīt login/password.

Projekts satur 3 svarīgākās mapes: app, src un web un failus root folderī, no kuriem būtu vērts pieminēt composer.json, kurā tiek glabāta informācija par sistēmai nepieciešamajām komponentēm. Uzstādot sistēmu, ērti izmantot composer rīku (<http://getcomposer.org/>), jo ar tā palīdzību tiek nolasīts composer.json fails un instalācijas gaitā tiek uzstādīti visas projektam nepieciešamās komponentes (pieļauju, ka šī varētu būt Maven analoģija priekš PHP). Composer izveido mapi vendor, kurā atrodas uzstādītās komponentes. Tā kā šī mape tiek automatizēti izveidota katrā projekta uzstādīšanas reizē, tā netiek izvietota repozitorijā. Turpinājumā īsumā pastāstīšu par augstāk minēto trīs mapju saturu.

### App mape

Šeit atrodas projekta konfigurācija. Symfony darbojas pēc principa, ka tiek veidotas *bundles*, kas ir kā savstarpēji neatkarīgi moduļi, kurus iespējams viegli pieslēgt kādam citam uz Symfony bāzētam projektam. App/AppKernel.php failā tiek reģistrēta *bundle* EKPDBundle, kas satur manis izstrādāto funkcionalitāti. App/config/config.yml tiek konfigurētas projekta komponentes, t.sk. pieslēgta monolog žurnalēšanas komponente. App/config/parametrs.yml satur informāciju par pieslēgšanos datu bāzei. Mape logs satur log failus. Tur atrodas arī error\_log.log un info\_log.log faili, kurus esmu izveidojusi priekš žurnalēšanas (tika norādīts config.yml failā).

### Src mape

Šeit atrodas projekta kods. Src/EK/PDBundle satur manis izveidotā *bundle* kodu. Controller mapē atrodas kontrolieri:

DefaultController.php – noklusētais kontrolieris;

AuthController.php – kontrolieris, kurš apstrādā lietotāju pieslēgšanos tīmekļa lapai;

UserController.php – kontrolieris, kurš atgriež informāciju par lietotāju;

Darbošanās dar datiem notiek pēc Object Relational Mapping (ORM) principa. Entity mapē atrodas klases, kuras tiek mappotas ar relāciju tabulām (no MVC viedokļa tie ir modeļi). Šobrīd datu bāzē tiek glabāta tikai informācija par lietotāju un klase tiek definēta User.php failā.

Resources mapē atrodas dažādi lapai nepieciešamie resursi. Ir vērts pieminēt mapi public un views. Mape public var saturēt js un css failus un attēlus, kuri tiek klonēti uz web mapi, lai no turienes tiktu izmantoti darbinot lapu. Public mapē šobrīd atrodas mape css ar style.css failu. Mape views satur lapas struktūru. Struktūru veidoju, izmantojot Twig (<http://twig.sensiolabs.org/>). Base.html.twig satur struktūras pamatu, kas ietver sevī head un citu visām tīmekļa lapām nepieciešamos elementus. Mapē Default atrodas index.html.twig, kurā ir sākumlapas struktūra (lapa, kurā ir facebook login). Mapē User atrodas fails view.html.twig, kurā ir lietotāja skats, kad viņš ir autorizējies.

Tests mapē atrodas Controller/DefaultControllerTest.php un Entity/UserTest.php faili, kuros atrodas manis rakstītie testi.

### Web mape

Šī ir web root mape. Te atrodas visi publiskie un statiskie faili, piemēram, css, js faili un attēli. Mapē atrodas arī front-kontrolieris. Pagaidām vienīgais fails, ko šajā mapē esmu mainījusi, ir style.css fails, kurš atrodas web/bundles/ekpd/css mapē. Tur šobrīd tiek definēts lapas noformējums.

## Paveiktais

Šobrīd lietotājs var pieteikties sistēmā, ievadot savu facebook lietotājvārdu un paroli. Ja lietotājs piesakās pirmoreiz, viņš tiek reģistrēts MySQL datu bāzē un tiek atvērta lapa, kurā viņš redz savu facebook profila bildi, vārdu un uzvārda pirmo burtu, epastu, dzimšanas datumu un vecumu. Tiek izvadīti kļūdu un informatīvie paziņojumi error\_log.log un info\_log.log failos. Ir izveidoti vienībtesti. Risinājums ir izvietots git repozitorijā.