

Teendők listája

- megírni a bevezetést - kész
- megírni a felhasználói doksit:
 - Röviden a "scrum"-ról - kész
 - Ismertetés -kész
 - Rendszerkövetelmények - kész
 - Telepítés - kész
 - Alkalmazás felépítése (képernyőképekkel):
 - * bejelentkező oldal - kész
 - * index oldal-navbar (siteadmin-user közti kül.) - kész
 - * projekt (lista, detail oldalak, funckiok) - kész
 - * epic, user story, task, issue - kész
 - * kanban (board, sttisztika) - kész
 - * munkaidő (könyvelés, timesheet, össznépi) - kész
- megírni a fejlesztői doksit:
 - Bevezető szöveg - kész
 - futtatás, supersuser létrehozás, konfigurálás - kész
 - Use case diagram - kész
 - alkalmazás felépítése (csomag diagramm): Adatbázis réteg + Szerver réteg + View réteg
 - Adatbázis: Postgresről, modellekről
 - Szerver: szolgáltatásosok, view-k

-
- View: templatek, használt CSS modulok/JS scriptek (pl.: chart)
 - tesztforgatókönyv



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

PROGRAMOZÁSELMÉLET ÉS SZOFTVERTECHNOLÓGIAI
TANSZÉK

Projektmenedzsment eszköz agilis módszertanhoz

Témavezető:

dr. Tejfel Máté

egyetemi docens

Szerző:

Nagy Levente

programtervező informatikus BSc

Budapest, 2020

Az eredeti szakdolgozati / diplomamunka témabejelentő helye.

Tartalomjegyzék

1. Bevezetés	2
2. Felhasználói dokumentáció	3
2.1. Röviden az agilis módszertanról és a "scrum"-ról	3
2.2. Rendszerkövetelmények	5
2.3. Telepítés	5
2.4. Az alkalmazás felépítése	6
2.4.1. Bejelentkezés, felhasználók	6
2.4.2. Főoldal	7
2.4.3. Projektek	8
2.4.4. Epic, Story, Task, Issue	9
2.4.5. Egy példa fejlesztési ciklus	10
2.4.6. Kanban tábla	11
2.4.7. Munkaidő napló	13
3. Fejlesztői dokumentáció	15
3.1. Konfiguráció	15
3.2. Funkcionális terv	16
3.3. Struktúrális felépítés	17

1. fejezet

Bevezetés

A mai világban egyre szélesebb körben alkalmazott az agilis módszertan az informatikai megoldásokkal foglalkozó cégeknél. A jelen dolgozat keretein belül elkészült alkalmazás (későbbiekben: ScrumHelper) az agilis módszertan szerint működő fejlesztői csapatok (azaz scrum-ok) segítésére született. Manapság sok, különböző ilyen eszköz elérhető, mind fizetős, mind nyílt-forráskódú változatban is. Ezek a projektek egyhamar hatalamsra nőnek, hogy minél több funkciót lássanak el, ezzel bonyolítva kezelésüket.

Ezen alkalmazás kisebb létszámú (például startup) cégek számára lehet előnyös elsősorban, de természetesen bármilyen scrum berendezkedésű fejlesztői csapatnak megfelelő. Az alkalmazást könnyen és egyértelműen lehet kezelni (részletesebb bemutatása felhasználók számára a 2. fejezetben olvasható).

A ScrumHelper a python programozási nyelv segítségével készült el, azon belül is a Django nyílt forráskódú keretrendszerrel, mely webes alkalmazások fejlesztésére szolgál, kihasználva a python nyelv adta sokdoldalú lehetőségeket. A teljesség igénye nélkül pár példa ezen előnyök közül:

- kevés soros és átlátható kódbázis segíti a gyorsabb fejlesztést
- jól dokumentált keretrendszer ¹
- modulokra (applikációkra) bontott szoftver segíti az újra felhasználhatóságot

¹<https://docs.djangoproject.com/en/3.0/>

2. fejezet

Felhasználói dokumentáció

Ezen fejezet a felhasználó részletes tájékoztatására szolgál. Az alfejezetek az alkalmazás szükséges előfeltételeit, telepítési és használati információkat tartalmaznak. A program technikai részletekbe menő dokumentációját a 3 . fejezet (Fejlesztői dokumentáció) tartalmazza.

2.1. Röviden az agilis módszertanról és a "scrum"-ról

Mi az agilis módszertan? A szoftverfejlesztési módszerek egy csoportja, ahol a követelmények és megoldások szoros együttműködésén keresztül fejlődnek az önszerveződő és multifunkcionális csapatok. Ez elősegíti a korai szállítást, folyamatos továbbfejlesztést és bátorít a változásokra adható gyors és rugalmas válaszokra. ²

Mi a "scrum"? Az agilis módszertanon belül sokféle irányzat van, ezek egyike a scrum. A scrum középpontjában a kis létszámú, önszerveződő agilis csapatok állnak. Itt, szemben a többi agilis ágazattal, nincsenek általánosan megszabott, egész rendszert egybefogó szállítási és frissítési időpontok, hanem az aktuális fejlesztések Sprintekbe rendeződnek és ezek lejártá után történik az átadás. Ez közvetlenebb visszajelzést biztosít a szállító és a megrendelő között. A scrum szerkezeti felépítése a következő: van egy scrum master, aki egybe fogja

²Részletes leírás megtalálható: <https://www.agilealliance.org/agile101/>

a csapat működését, feladata a scrum "menedzselése" gyakorlatilag. Mellette a csapat tartalmaz természetesen fejlesztőket és tesztelőket. A scrum létszáma nincsen hivatalosan meghatározva, de ajánlatos 8-10 főnél nem nagyobbra nőnie, ugyanis így veszít hatékonyságából. A csapatok minden nap egy meghatározott időpontban tartanak stand up-okat, amelyeken minden tag beszámol feladatairól, haladásáról, így mindenki a csapaton belül képbe kerülhet a Sprint aktuális állásáról.

Pár fontosabb fogalom, amely a későbbiekben külön fejezetben részletezve lesz:

- Projekt 2.4.3. fejezet
- Epic, User stories, Task, Issue 2.4.4. fejezet
- Kanban 2.4.6. fejezet

2.2. Rendszerkövetelmények

Minimum követelmények: A ScumHelper egy webes alkalmazás. Ez azt jelenti, hogy a felhasználónak csak egy böngészőre van szüksége a számítógépén (például: Google Chrome, Mozilla Firefox, Safari, stb.) és természetesen internet elérésre ahhoz, hogy futtatni tudja a szoftvert. Utóbbi elengedhetetlen, ugyanis csak bejelentkezve lehetséges használni, továbbá internet kapcsolat nélkül a számítógép csak a gyorsítótárában mentett oldalakat tudja megnyitni, de módosításokat nem tudunk végrehajtani a betöltött oldalon és új lapot sem tudunk megnyitni az alkalmazáson belül. Nincsen megkötés operációs rendszer tekintetében, tehát minden, napjainkban használatos rendszeren (Linux, Windows, Mac OS) egyaránt használható.

Ajánlott követelmények: A jobb felhasználói élmény érdekében érdemes legalább 1280x720 felbontású kijelzőn használni és a korábban már felsorolt, jelenleg leggyorsabbnak és legbiztonságosabbnak számító böngészővel megnyitni : Chrome, Mozilla firefox, Chromium, valamint érdemes szélessávú internet eléréssel rendelkezni.

2.3. Telepítés

A felhasználói oldalról nem igényel telepítést. Az eléréshez a szerver elérési URL-jére van szükség, illetve egy felhasználó igénylésére az aktuális rendszergazdától. Érdemes az új felhasználóba való belépés után megváltoztatni jelszavunkat biztonsági okokból.

Rendszergazdai oldalról ha még nem rendelkezik felhasználóval, akkor a fejlesztői dokumentációban (3. fejezet) található információ a rendszergazda felhasználó létrehozásának lépéseiről. Ha futtatni szeretné az alkalmazás szerverét lokálisan a saját számítógépen, vagy egy kihelyezett szervergépen, akkor azon rendszeren telepíteni kell a következő programokat: python (3.5 vagy később verzió) ³, django és django-extension python csomagok⁴, valamint egy adatbázis kezelő szoftvert (PostgreSQL, Oracle, MySQL, MariaDB, SQLite).

³<https://www.python.org/downloads/>

⁴<https://docs.djangoproject.com/en/3.0/intro/install/>

2.4. Az alkalmazás felépítése

2.4.1. Bejelentkezés, felhasználók

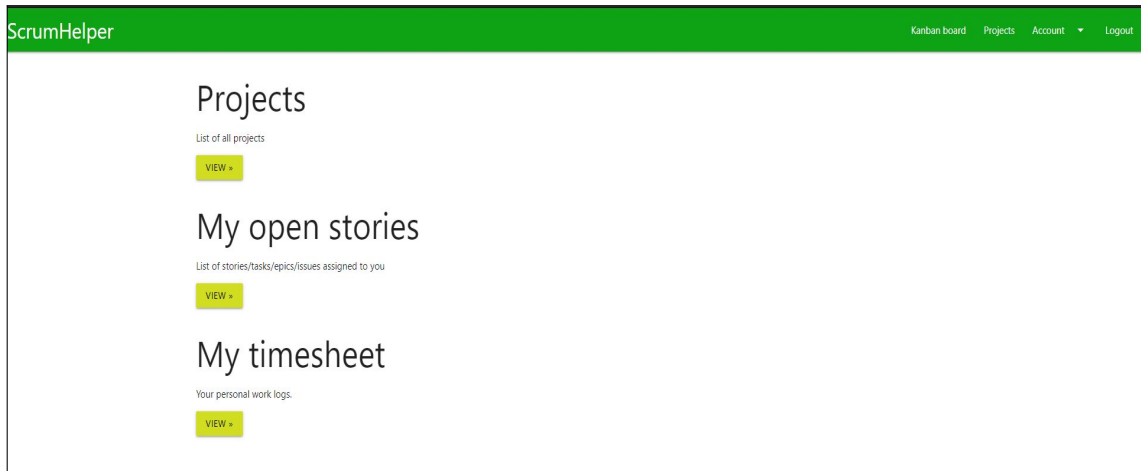
Az alkalmazásban az első képernyő, amely fogadja a felhasználót az a login oldal. A ScrumHelper-t csak sikeres bejelentkezés után lehetséges használni. Ha nem rendelkezik felhasználóval, akkor keresse meg a rendszergazdát, és igényeljen egyet. Az egyes felhasználói fiókok egy-egy jogosultsági csoporthoz kötöttek.

A felhasználói csoportok és jogosultságaik:

Csoport	Jogosultságok
<i>Fejlesztő</i>	projekt/epic/story/task/issue létrehozás/szerkeztés, komment létrehozás/törlés, saját munkaidő napló létrehozása/törlése, Dokumentum feltöltés (ha projekt tulajdonos, akkor törlés), jelszóváltás
<i>Tesztelő</i>	projekt/epic/story/task/issue létrehozás/szerkeztés, komment létrehozás/törlés, saját munkaidő napló létrehozása/törlése, Dokumentum feltöltés (ha projekt tulajdonos, akkor törlés), jelszóváltás
<i>Scrum master</i>	projekt/epic/story/task/issue létrehozás/szerkeztés/-törlés, komment létrehozás/törlés, saját munkaidő napló létrehozása/törlése/egész scrum könyvelés megjelenítése, Dokumentum feltöltés (ha projekt tulajdonos, akkor törlés), jelszóváltás
<i>Projekt menedzser</i>	projekt/epic/story/task/issue létrehozás/szerkeztés/-törlés, komment létrehozás/törlés, saját munkaidő napló létrehozása/törlése/egész scrum könyvelés megjelenítése, Dokumentum feltöltés (ha projekt tulajdonos, akkor törlés, jelszóváltás)
<i>Rendszergazda</i>	minden jogosultsággal rendelkezik, létre is hozhat új csoportokat, kezelheti azok jogosultságait, illetve a felhasználókat is tudja szerkeszteni/törölni

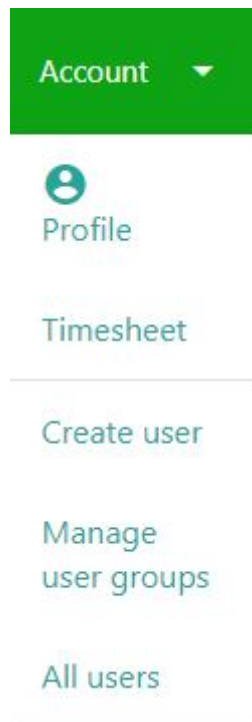
2.1. táblázat. Jogosultsági csoportok és jogosultságaik

2.4.2. Főoldal



2.1. ábra. Főoldal

A 2.1. ábrán látható a főoldal kinézete. A navigációs sávban az alkalmazás neve, Kanban board, Projects, Account és Logout mezőket olvashatjuk. A kanban board el-navigál a scrum Kanban táblájára, ahol az összes felvett feladatot láthatjuk egyben. A Projects menüponttal juthatunk el az adatbázisban szereplő projektek listájához. Az Account menüpont egy legördülő menü, amelyben a 2.2. ábrán látható tartalom jelenik meg.



2.2. ábra. Account legördülő menü

Az elválasztó vonal alatti almenük: Create user, Manage user groups, All users csak a rendszergazda jogosultságú felhasználók számára láthatóak. A Logout menüpont értelemszerűen kijelentkezteti a felhasználót.

A főoldal törzsében található 3 opció: Projects - a Projects menüponttal megegyezően a projektek listájára navigál, My open stories - a bejelentkezett felhasználóhoz rendelt story-k/task-ok/issue-k tekinthetőek meg és a profil adatok, My timesheet - a felhasználó munkaidő könyvelési oldalára kalauzol.

2.4.3. Projektek

A projektek listájához a Projects menüpont, illetve a főoldalról tudunk eljutni. Az oldalon egyszerre 5 projekt jelenik meg, a többbit a 2.3.ábra alján látható oldal léptetéssel érhetjük el. A projektek létrehozási dátum szerint a legújabbtól haladva a legrégebbi felé jelennek meg. A projekt oldalak esetében a navigációs sávban is megjelenik egy "Create project" nevezetű gomb, mellyel egyből a létrehozás oldalra juthatunk.



2.3. ábra. Projektek listája

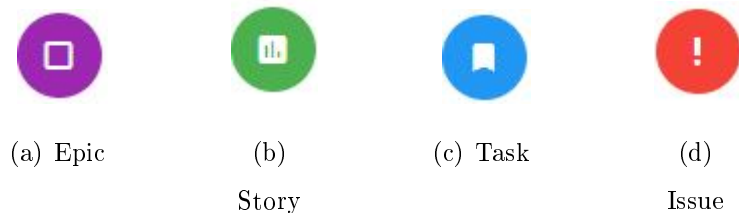
Az egyes projektekre kattintva eljuthatunk a projekt saját oldalára. Itt megtalálhatjuk az alapvető információkat a projektről, a projektekhez feltöltött dokumentumokat, valamint a projekthez tartozó összes feladat (epic,story,task,issue) felosolását. A jobb alsó sarokban található egy menü, mely az egyes feladat fajták

saját oldalain is megtalálhatóak, természetesen részben eltérő menüpontokkal. A projektek esetében ez tartalmazza a szerkeztés, epic, story, task, issue létrehozását, valamint a projekt szerkeztését, illetve a megfelelő jogosultsággal rendelkezőknek a törlés opciót. (2.4. ábra).



2.4. ábra. almenü, mely különböző opciókat tartalmaz projekt, vagy feladat típustól függően

2.4.4. Epic, Story, Task, Issue



2.5. ábra. Az egyes feladat típusok ikonjai

Epic: Egy projekten belül levő feladatok gyűjteménye. Mivel projektek több sprinten át élhetnek, ezért rövidebb időszakokban az összefüggő story-k és taskok összefogására érdemes ezt használni. Létrehozni az adott projekt oldalán a korábban említett almenüből lehet (2.4. ábra).

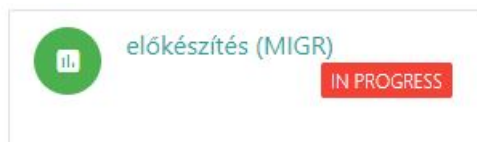
Story: Felhasználói feladat. Ha valamilyen jól körülhatárolt fejlesztés van a projekten belül, akkor érdemes ezt használni annak leírására. Létrehozatalakor még nem kötelező hozzárendelni felhasználót, lehet később is a story szerkeztésével. Létrehozni a projekt oldalán levő almenüből (2.4. ábra) lehetséges, avagy a navigációs sávban megjelenő "Create story" gombra kattintva. Ez olyankor elérhető, amikor egy meglévő story oldalán vagyunk éppen.

Task: Szintén felhasználói feladat. A taskot a storyval szemben kisebb feladatok leírására érdemes használni. Például, ha valaki dokumentációt ír, vagy mee-

tingekre jár, vagy csak valami apró fejlesztésről van szó, akkor érdemes azt "taskosítani". Létrehozni a projektek, illetve epicek és storyk oldalán lehetséges az almenüből (2.4. ábra).

Issue: Az issue kicsit eltér a storytól és a tasktól. Az issue valamilyen jellegű probléma leírására használható. Például ha egy elkészült fejlesztésben a tesztelő hibát talál, vagy ha már a leszállított build-ben derül ki valamiféle "bug". Létrehozni a projekt almenüjében lehetséges (2.4. ábra).

A story, task és issue hármas státusszal is rendelkezik. Az aktuális státuszt minden olyan oldalon láthatjuk, ahol valamilyen felsorolásban szerepel a 3 közül bármely feladat típus. A státuszt az adott feladat saját oldalán lehet a jobb felső sarokba látható gombbal állítani, melyen mindig az a státusz olvasható, amely az éppen aktuálisat követi.



2.6. ábra. Példa egy story-ra egy felsorolásban: látható a neve, a projekt kódja, emblémája és aktuális státusza

2.4.5. Egy példa fejlesztési ciklus

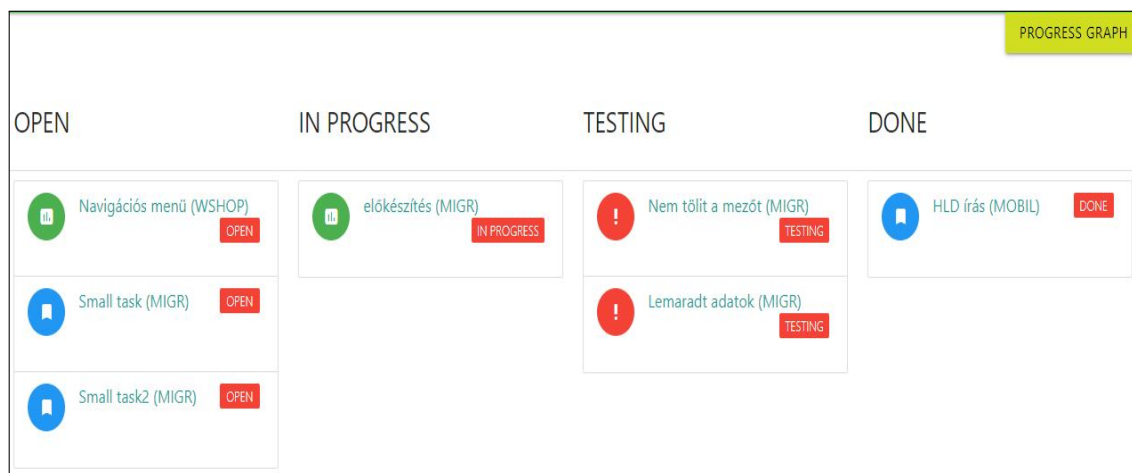
Az egyes fogalmak gyorsabb és könnyebb megértése érdekében egy példa fejlesztési folyamaton keresztül szemléltetném a használatukat:

1. A scrum master-hez megérkezik a fejlesztési igény (projekt) a kientstől.
2. Felvesz egy projektet a ScrumHelper-ben. Feltölti hozzá az igényhez kapott dokumentumokat, amelyek segítik majd a tervezők munkáját.
3. Ezt követően kiosztja az igény megtervezését egy tervezőnek (architect). Ehhez task-ot készít, ahol összeírja a teendőket röviden.
4. A tervező megírja az igény specifikációját arról, hogyan lehetne ezt az adott rendszerben implementálni (High Level Solution Design).

5. Utóbbi feltölti a projekt többi dokumentuma közé, majd elkezd lebontani feladatcsoportokra (epic) és feladatokra (story).
6. A fejlesztő ezután válogathat a story-k között, melyiket tudja megcsinálni. Amelyiket kiválasztja azt magához rendeli a story szerkeztői oldalán és IN PROGRESS-be rakja a státuszt.
7. Amint végzett a fejlesztéssel, TESTING státuszba állítja. Egy tesztelő megke-
resi (például a Kanban boardon) és magához rendeli.
8. Ha tesztelés közben hibát talál, akkor felvesz a projekthez egy Issue-t amit az
eredeti fejlesztőhöz (vagy aki javítani tudja) rendeli.
9. Amikor már nincsen probléma a story-val, akkor DONE státuszba rakja.
10. Egy feladat életciklusának utolsó állomása a CLOSED, amelyre a sprint végén
ajánlatos állítani. Ha valami okból újra foglalkozni kell vele, akkor lehetőség
van a REOPEN opcióval újra nyitottá tenni.
11. Egyes Release-k idején, hogy ne legyen túl sok a story, érdemes a rendszergaz-
dáknak egy karbantartó scripttel kiüríteni a már régebben lezárt feladatokat
(részletek a 3. fejezetben).

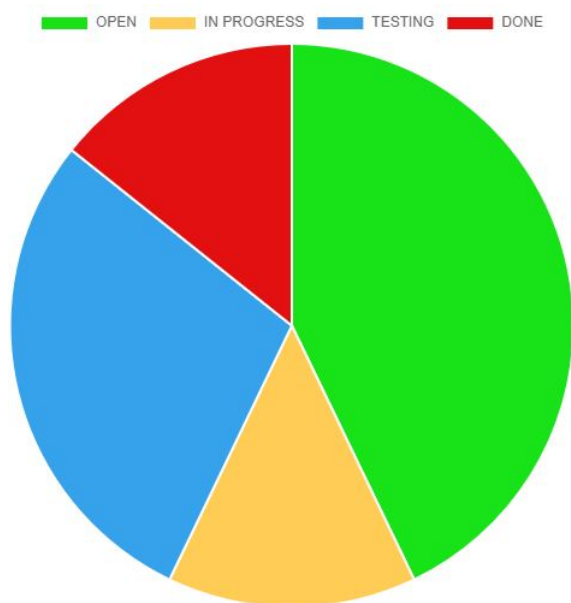
2.4.6. Kanban tábla

A kanban szintén egy egységesen használt fogalom (avagy eszköz) a scrum mód-
szertanban. A kanban board - azaz "tábla" - arra szolgál, hogy egybe gyűjtse a scrum
összes felvett feladat típusát. A ScrumHelper-ben ez a funkció a navigációs sávban
a Kanban board menüpontra kattintva érhető el. Itt négy státusz alapján négy
oszlopba osztva láthatóak az aktuális feladatok: OPEN (nyitott), IN PROGRESS
(folyamatban), TESTING (tesztelés alatt), DONE (kész).



2.7. ábra. Kanban tábla

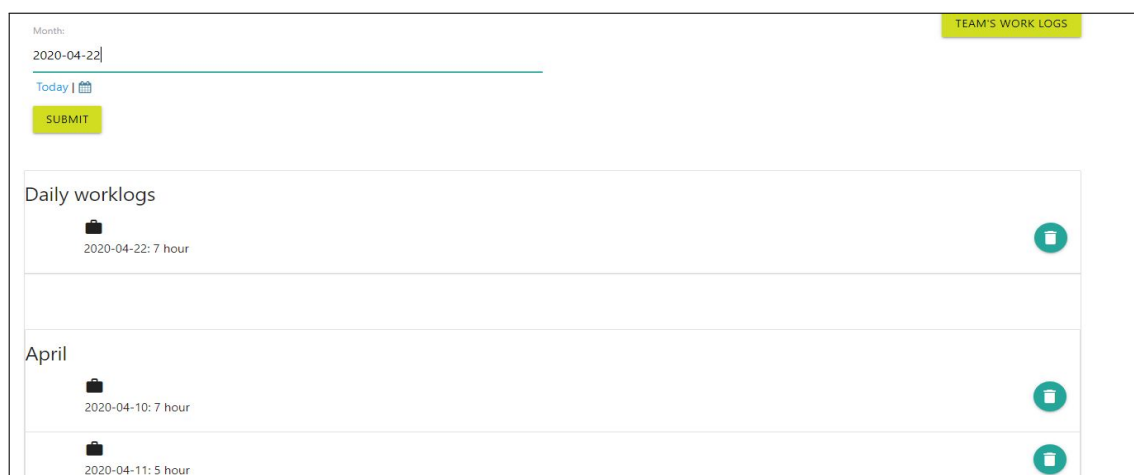
A jobb felső sarokban látható gomb (2.7. ábra "Progress graph") megnyomásával egy körgrafikon segít az aktuális feladatok állásának nyomonkövetésében. Az alábbi ábrán egy példa:



2.8. ábra. Körgrafikon az aktuális feladatok jelen státuszáról

2.4.7. Munkaidő napló

Az alkalmazás lehetőséget biztosít az egyes feladatokkal eltöltött munkaidő számontartására is. Ehhez el kell navigálnunk az adott feladat (story, task, issue) oldalára és a már korábban többször is említett (2.4. ábra) jobb alsó sarokban található almenüben kiválasztani a munkaidő könyvelése opciót ("Add worklog", ikon: kék körben fehér aktatáska). Ez eljuttatja a felhasználót a munkaidő létrehozó oldalára. Itt a dátum mezőt megfelelő formátumban kitöltve, avagy a naptár ikont használva meg kell adni a munkanapot. Alatta pedig megadni a feladattal töltött munkaórák számát, amely 1 és 8 közé kell essen. Ettől eltérő bementre figyelmeztet az alkalmazás, hogy rossz adatot adott meg. A navigációs sáv "Account" legördülő menüjében (2.2. ábra) a "Timesheet" opcióra kattintva, avagy a főoldalról a "Timesheet" pont-ra kattintva tekinthető meg a saját profilunkhoz rögzített összes munkaidő napló bejegyzés napi és havi bontásban (2.9. ábra).



2.9. ábra. Személyes munkaidő napló

Kiválasztható a dátum napi pontossággal, akár kézzel beírva a megfelelő formátumban, akár a naptár ikonra kattintva a felugró naptárból. A jobb felső sarokban látható egy "TEAM'S WORK LOGS" feliratú gomb. Erre kattintva megtekinthető az egész scrum havi munkaidő könyvelése (2.10. ábra) Az egyéni naplóhoz hasonlóan, egy megadott dátum alapján az adott hónapra megjeleníti az egyes felhasználók által lekönyvelt össz óraszámát. A jogosultságoknál említettek alapján, csak a scrum vezetők, a projektmenedzserek és a rendszergazdák látják ténylegesen minden csapattag óraszámát. A többi csoport csak a sajátját látja összegezve. Törölni

bejegyzést minden felhasználó csak maga tud, nem lehet másét eltávolítani. Ezt a személyes munkaidő napló oldalon lehet megtenni az egyes bejegyzés mellett kuka ikonra kattintva.

Month:		PERSONAL WORK LOGS
<div>SUBMIT</div>		
User	Work hours	
dummydev1	0	
dummyPM	14	
dummytester	8	
levi	30	
dummyScrumMaster	16	

2.10. ábra. Scrum felhasználóinak össz munkaórája az adott hónapban (amit a scrum vezető lát)

3. fejezet

Fejlesztői dokumentáció

Ez a fejezet fejlesztőknek nyújt segítséget a ScrumHelper feltérképezésében. Az alfejezetek kifejtik az alkalmazás felépítését, részletesen leírják a különböző rétegeknek (Adatbázis-Szerver-Nézet) osztályait és függvényeit. A vizuális segítséghez különböző diagrammokat is tartalmaz (osztály-, csomag-, felhasználói esetek diagram). A fejezet végén egy tesztforgatókönyv ad részletes leírást a teszt esetek és azok eredményeiről.

3.1. Konfiguráció

A futtatáshoz szükséges előkövetelmények a felhasználói dokumentáció 2.3. alfejezetében olvashatók. A dolgozott keretein belül csak a lokális szerveren való konfigurálás és futtatás lesz részletezve. A különböző szervergépeken való futtatáshoz részletesebb információt a hivatalos Django dokumentációban lehet találni. Utóbbi eset külön konfigurációt igényel a wsgi.py, illetve asgi.py file-ok és környezeti változók megfelelő beállításának segítségével ⁵.

Ahhoz, hogy lokálisan futtatni tudjuk a szervert, először is a ScrumHelper/ScrumHelper/setting.py file-ban kell beállítanunk a DATABASES változót. A pontos beállításai eltérőek a különböző adatbázisok esetében, ehhez részletes segítséget nyújt a hivatalos Django dokumentáció. Alább egy PostgreSQL adatbázis konfigurációja látható:

⁵wsgi szerver: <https://docs.djangoproject.com/en/3.0/howto/deployment/wsgi/> asgi szerver: <https://docs.djangoproject.com/en/3.0/howto/deployment/asgi/>

```
1
2 DATABASES = {
3     'default': {
4         'ENGINE': 'django.db.backends.postgresql',
5         'NAME': 'database name',
6         'USER': 'database user',
7         'PASSWORD': 'password',
8         'HOST': '127.0.0.1',
9         'PORT': '5432',
10    }
11 }
```

3.1. forráskód. Konfiguráció PostgreSQL adatbázis használatához

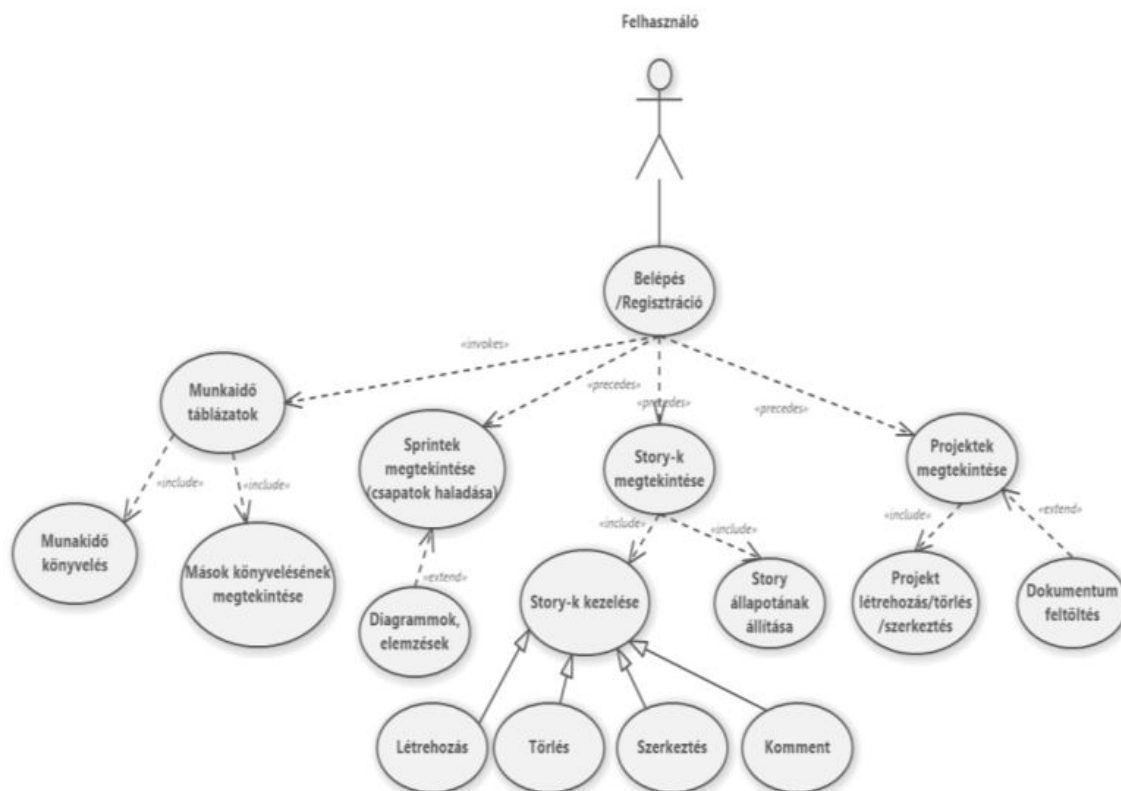
Ha ez megfelelően van beállítva, akkor a ScrumHelper fő mappába navigálva kell lefuttatni a "python manage.py migrate" parancsot minden első futtatásnál a szükséges adatbázis struktúra kialakításához (illetve ha fejlesztés során olyasmi változik, amely érinti az adatbázis struktúrát, akkor a makemigrations-t is le kell a migrate parancs előtt futtatni). Ezután a "python manage.py runserver" elindítja a lokális szervert a localhost:8000-es portján. Superuser-t, azaz minden jogosultsággal rendelkező felhasználót a szerver futtatása nélkül lehet generálni: "python manage.py createsuperuser" paranccsal, megadva a felhasználónevet és jelszavat utána.

A TIMEZONE változóban adható meg az időzóna. Az egyes projektekhez feltöltött fájlokat a MEDIA_ROOT konfigurációs változóbeli elérési úton elhelyezkedő mappába menti (ez alpból a ScrumHelper/media könyvtárra mutat). Ugyanezen az elven kezeli a statikus fájlokat is a keretrendszer, ennek a környezeti változója a STATIC_ROOT.

Fontos, hogy mielőtt az alkalmazás valós, produkciós használatba kerülne szükséges a DEBUG változót False, azaz hamisra állítani (biztonsági okokból).

3.2. Funkcionális terv

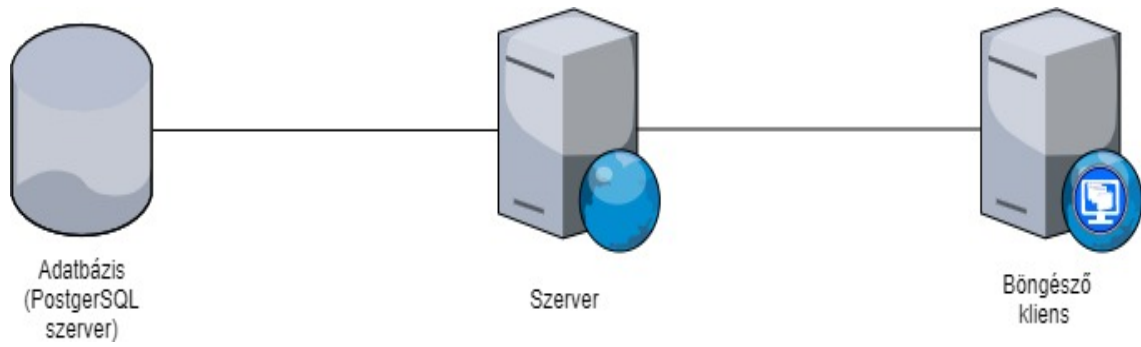
Az 3.1. ábrán egy felhasználói eset diagram mutatja be az alkalmazás funkcióit:



3.1. ábra. Felhasználói eset diagram

3.3. Struktúrális felépítés

A ScrumHelper egy MVC (Model-View-Controller) alkalmazás, bár a hivatalos Django dokumentációban az MTV (Model-Template-View) kifejezést használják, mivel ezt gondolják pontosabb leírásának. Tekintve, hogy Djangóban írodott az alkalmazás, ezért ebben a dokumentációban is utóbbi analógia a mérvadó. Ez azt jelenti, hogy három rétegből tevődik össze: egy Model rétegből, amely az adatbázist hivatott reprezentálni, egy View rétegből, amely az adatot reprezentálja (ez alatt azt értjük, amit az adatbázisból kigyűjtött, nem feltétlenül a felhasználó számára megjelenített adatot), valamint a különböző template-ek renderelését is végzi, továbbá egy Template rétegből, mely leírja, hogyan legyen az adat megjelenítve a felhasználó számára.



3.2. ábra. Az alkalmazás rétegei: egy adatbázis, egy szerver, ami kinyeri belőle az adatot és továbbítja a nézetnek, valamint a nézet amit a böngészőben láthatunk.