

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)
 - * munkaidő (könyvelés, timesheet, össznépi)
- megírni a fejlesztői doksit:



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 könyvelés	11

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 lélekszámú (például startup) cégek számára lehet előnyös elsősorban, de természetesen bármilyen scrum berendezkedésű fejlesztői csapatnak megfelelő. Az alkalmazást könnyen és egyértelműen lehet kezelni (részletesebb bemutatása 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 ?? . 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 az korai szállítást, folytonos 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 is biztosít a szállító és a kliens között. A scrum szerkezeti felépítése a következő: van egy scrum master, aki egybe fog-

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

ja 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 2.4.4. fejezet
- User stories 2.4.4. fejezet
- Task 2.4.4. fejezet
- Issue 2.4.4. fejezet
- Kanban ??. 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 gyorsírótarában mentett oldalakat tudja megnyitni, de módosításokat nem tudunk végrehajtani az oldalon és váltani sem tudunk az alkalmazáson belül. Nincsen megkötés operációs rendszer tekintetében sem, 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.

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 (?? fejezet) található részletesebb információ az alkalmazás beüzemeléséről, illetve rendszergazdai felhasználó létrehozásának lépéseiről.

2.4. Az alkalmazás felépítése

2.4.1. Bejelentkezés, felhasználók

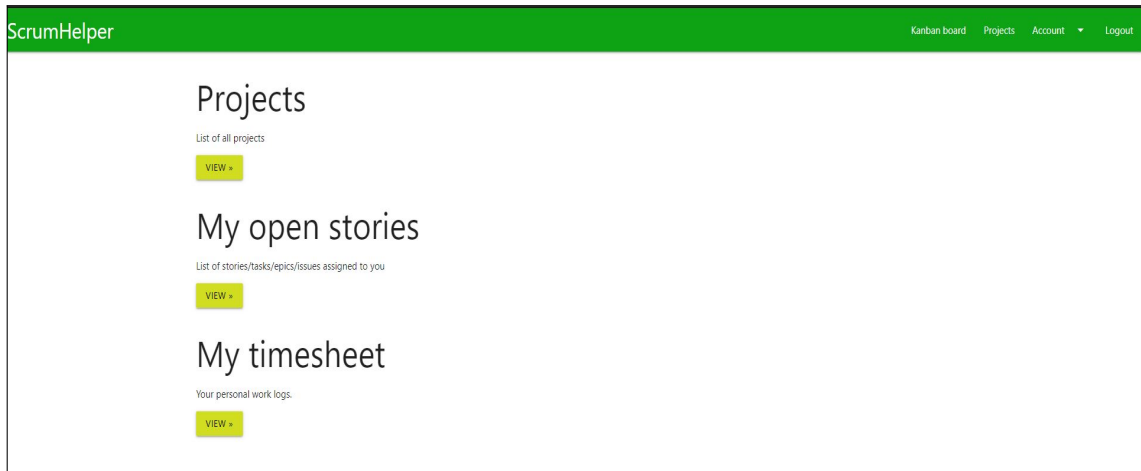
Az alkalmazásban az első képernyő, amely fogadja a felhasználót az a login oldal. A ScrumHelper-t csak sikeres bejelentkezés után lehetséges használni. Ha nem rendelkezik felhasználóval, akkor keresse meg a rendszergazdát, és igényeljen egyet. Az egyes felhasználói fiókok egy-egy jogosultsági csoporthoz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

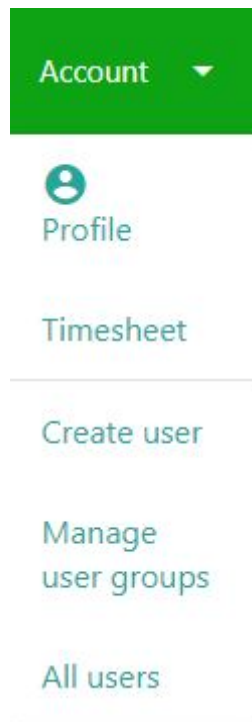
2.1. táblázat. Jogosultsági csoportok és jogosultságaik

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.



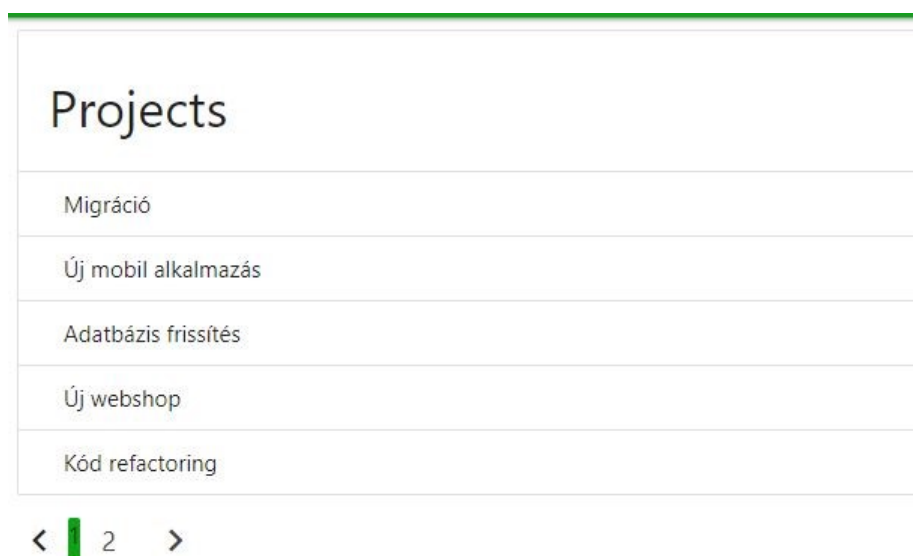
2.2. ábra. Account legördülő menü

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 tekinthetők 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 többi 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 ezt 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 emblémái

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 menün 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 (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 kientől.
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 ?? fejezetben).

2.4.6. Kanban tábla

2.4.7. Munkaidő könyvelés