

Online Project Management

csapatnév: OPM

Csapattagok

Papós Zita	CL5MAE	papos.zita@gmail.com	Mobil / Android Studio(Kotlin)
Bui Quang Anh Krisztián	SU4NX2	bui.krisztian01@gmail.com	Backend / Node.js, JWT, REST API
Fischer Balázs	D3MKVM	fischerbalazs2002@gmail.com	Frontend / Angular, Bootstrap, CSS
Imets Ákos István	D7H8G6	imets01@gmail.com	Frontend / Angular, Bootstrap, CSS,

Specifikáció:

Az alkalmazás projektek menedzselését és annak állapotának nyilvántartását teszi lehetővé webes felületen vagy Android alkalmazáson keresztül. A program Node.js futtató környezetet és AngularJS keretrendszert fog használni.

Funkciók:

- Szerepkörök kiosztása:
 - Admin
 - Projectmanager
 - Project tagok
- Projekt létrehozása
- Projektben:
 - Új task hozzáadása
 - Emberhozzárendelés
 - Határidő
 - Állapot
 - Mérföldkő
- Projekt befejezése, archiválás

Use casek:

- Admin:
 - Projekt létrehozás, manager és tagok kijelölése
 - Minden létező projektet, taskot, milestone-t lát
 - Projektben belül:
 - Név megadása
 - határidő beállítása,
 - archiválás,
 - törlés
 - Taskon belül:
 - felvétel,
 - név módosítás,
 - állapot módosítás,
 - határidő módosítás,
 - Workerek módosítása
 - egymásra épülő taskok megadása
 - Leírás módosítás
 - Archiválás
 - Törlés
- Projectmanager:
 - Azon projekteket látja melyhez hozzá van rendelve az admin által
 - Ő irányítja a projektben belüli taskok minden állapotát:
 - felvétel,
 - név módosítás,
 - állapot módosítás,
 - határidő módosítás,
 - Workerek módosítása

- egymásra épülő taskok megadása
 - Leírás módosítás
 - Archiválás
 - Törlés
- Project tag:
 - Azon projekteket látja melyhez hozzá van rendelve az admin által
 - A projekten belüli taskokat, milestoneokat
 - Nem tud módosítást végezni ezeken

Bejelentkezés után láthatóak az adott felhasználó típusához kapcsolódó elemek:

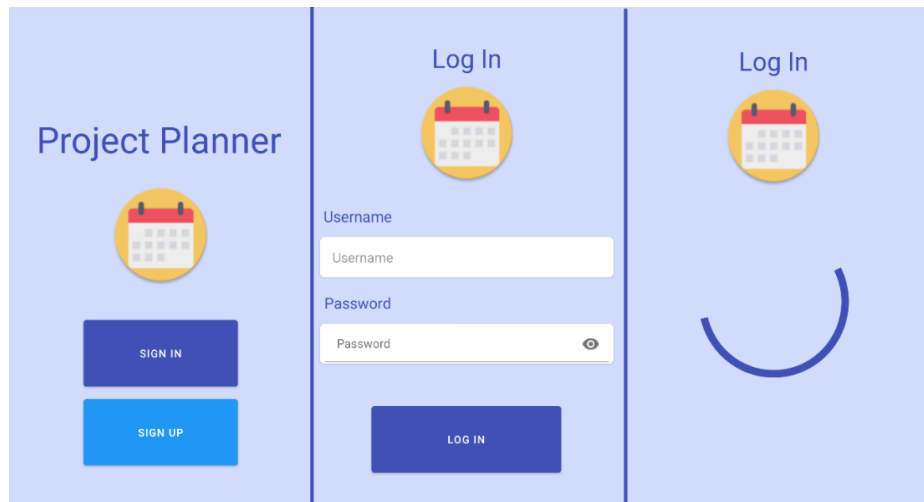
- Admin: Minden projekt, task, milestone
- Projectmanager: Az adott project melyet irányít, minden hozzá tartozó task és milestone
- Project tag: Azon projectek melyekhez fel lett véve, ezen belüli taskok és milestone

Mobil:

Az MVVM architektúra kiépítése, és a hálózati hívások megvalósítása:

Az architektúra hatékony felhasználásához szükség volt Retrofit kliensre, Network DataSourceokra, API osztályokra, data class-ekre, Repositorykra, ViewModelekre, illetve ViewState osztályokra is. Ez első ránézésre bonyolultnak tűnt, de a kapott videók alapján már logikussá vált.

A hálózati elérés is okozott kisebb bonyodalmakat, főleg a Login AccessToken, illetve RefreshToken használata. A RefreshToken-t az API-tól ugyanis Cookie-ként kapjuk meg, ennek használatáról pedig nem igazán tudtam sokat, de internetes tutorialok alapján ezt is implementálni tudtam. Így a bejelentkezés már működött.



1. Bejelentkezés folyamata

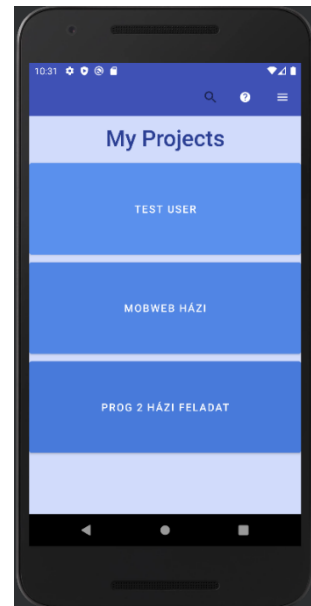
Érdekes részlet volt továbbá az infinite scroll megvalósítása is. Ezt úgy használtam, hogy a project-ek, milestone-ok, task-ek listájának lekérésekor mindig csak 5db-ot kér le a szervertől, majd mikor a felhasználó ennek az aljára ér, akkor kérem le a következő 5öt. A toolbaron a keresés gomb a projektek között keres név alapján.



3. Nézet választó

Ezután a projektekre kattintva egy olyan fragmensre irányít az alkalmazás, ahol kiválaszthatjuk, hogy a taskeket milestoneokba rendezve szeretnénk látni, vagy az összeset 1 helyen. Itt a következő felsorolások a projekt lapéhoz hasonlóan infinite scrollal működnek.

A toolbarban, ha már egy projektválasztási lépésen túl vagyunk, megjelenik egy ikon, amire kattintva megkapjuk az aktuális projekt információit. Ezt szintén hálózati lekérdezéssel.

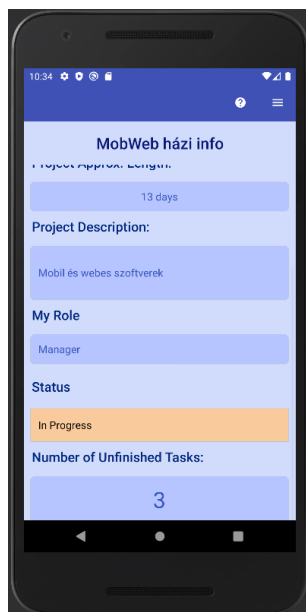


2. Projects lap

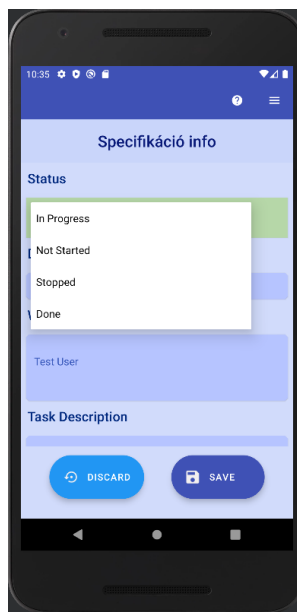
Ha előzmény feladat nincs befejezve, azt egy listán láthatjuk, egy piros ikon jelzi a task neve mellett. Az előzmény feladatokra kattintva az alkalmazás felhossa annak információit. (a screenshotok a következő oldalon)



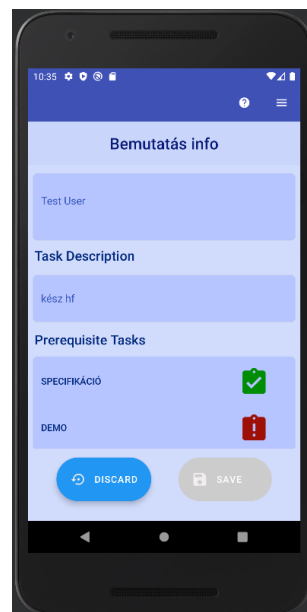
4. Task lista



5. Projekt információk



6. Task status váltás

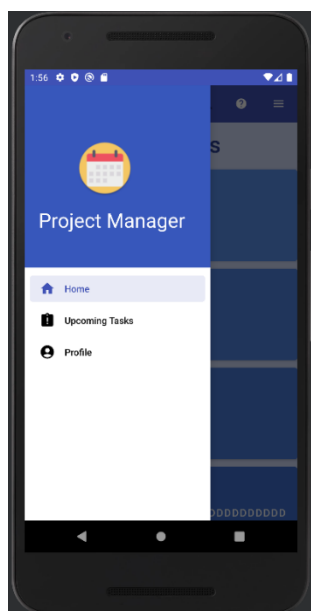


7. Befejezetlen előzmény task

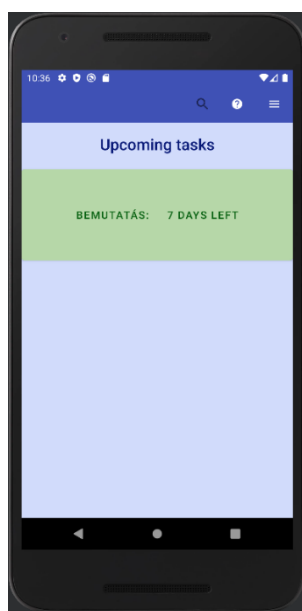
Az alkalmazás különböző funkciói közötti váltást egy NavigationView-val oldottam meg. Itt lehet váltani a projekt nézet, az Upcoming Tasks, illetve a Profile lap között. Ennek mindnek észlelhetően különbözik a funkciója, ezért külön kezeltem.

Az Upcoming Task lapon a felhasználó megkapja azokat a taskeket, melyeknek határideje a következő 2 hétben van, és még nincsenek befejezve. Itt a határidő közelsége alapján színekódolva vannak.

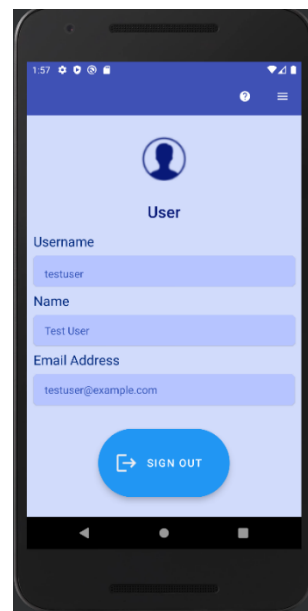
A Profile lap magától érthetően a szerverről lekérdezi az éppen bejelentkezett profil adatait, majd ezt mutatja. Itt lehet kijelentkezni.



8. NavigationView



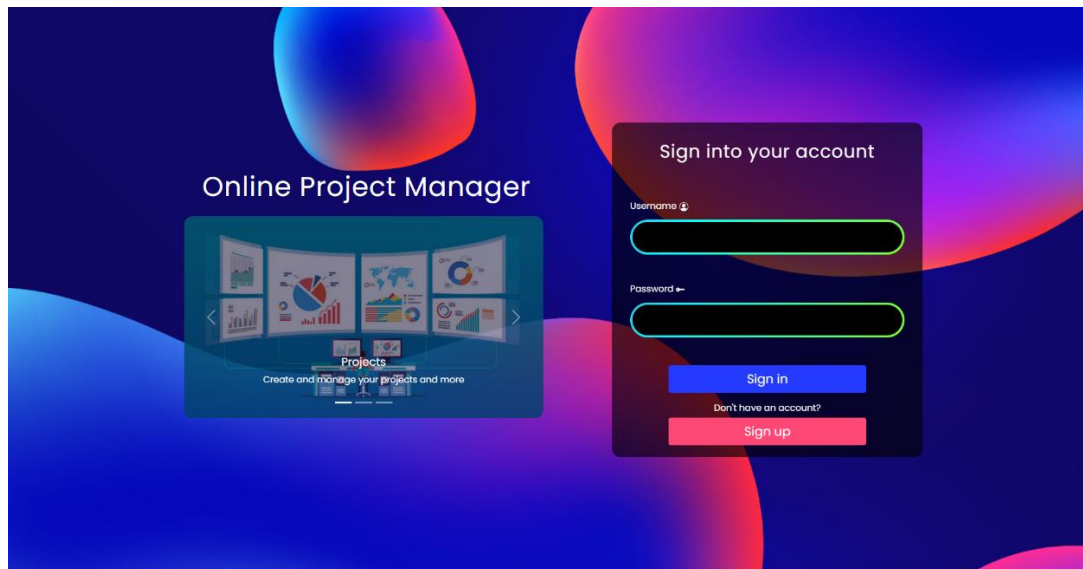
9. Upcoming Tasks lap



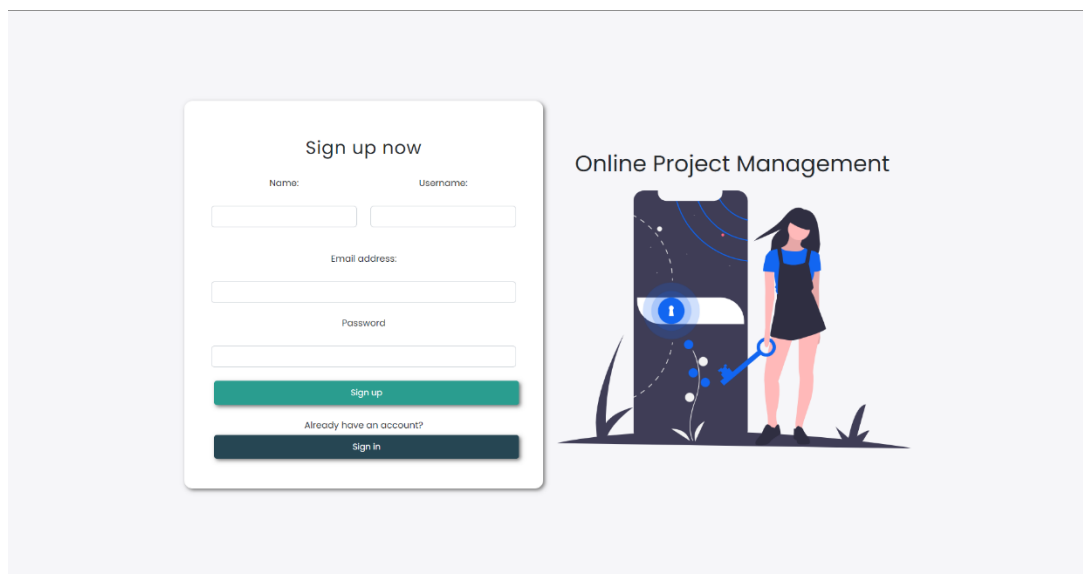
10. Profile lap

Web:

A webes felületet elindítva a sign in nézet fogadja a felhasználót, itt lehetőség van már meglévő felhasználó esetén bejelentkezésre, új felhasználó esetén a sign up lehetőséggel regisztrálásra.

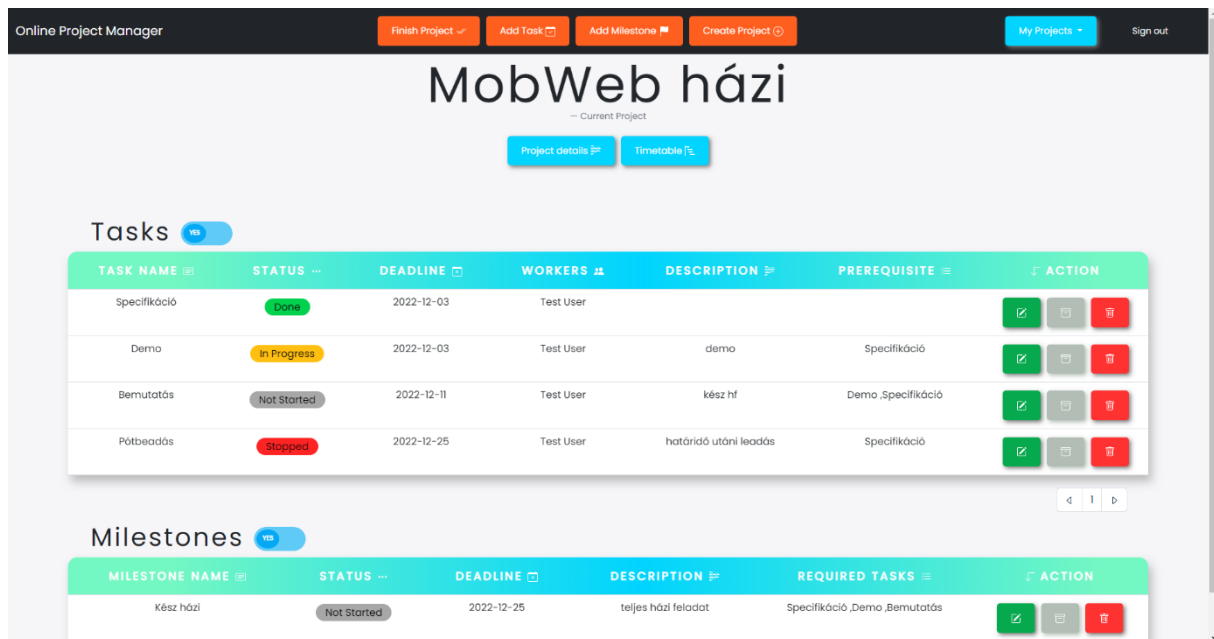


1. Sign in nézet



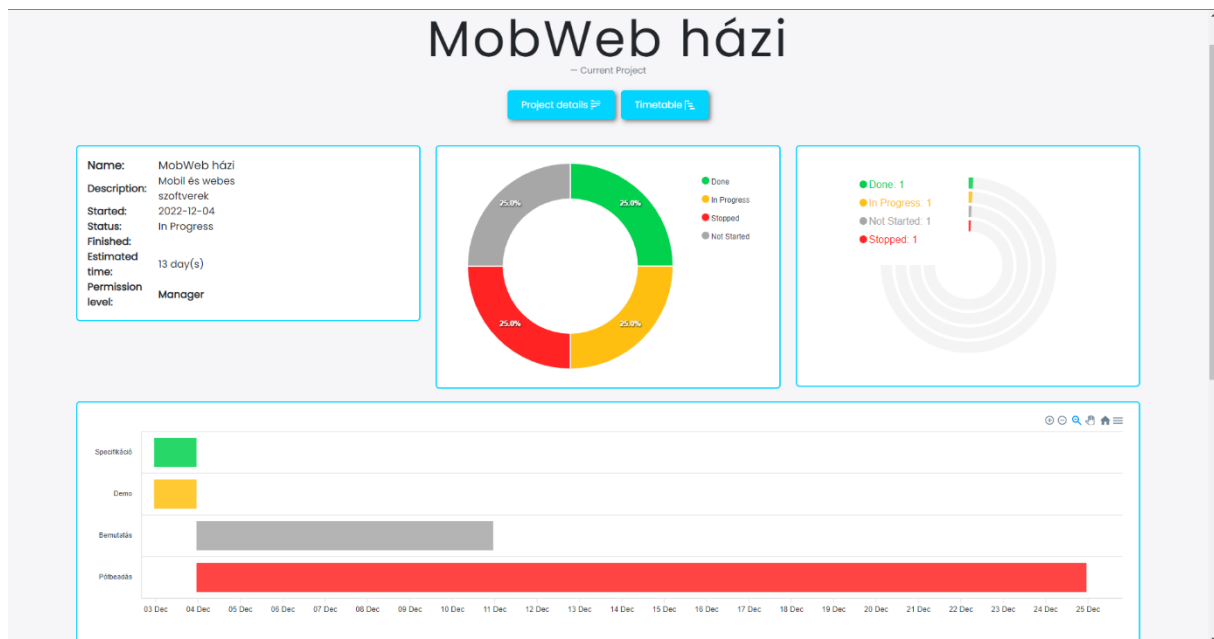
2. Sign up nézet

Sign up esetén a szokásos regisztrációs adatokat megadva jön létre az új felhasználó, az Access Token segítségével fog hozzá férni a weboldalhoz.

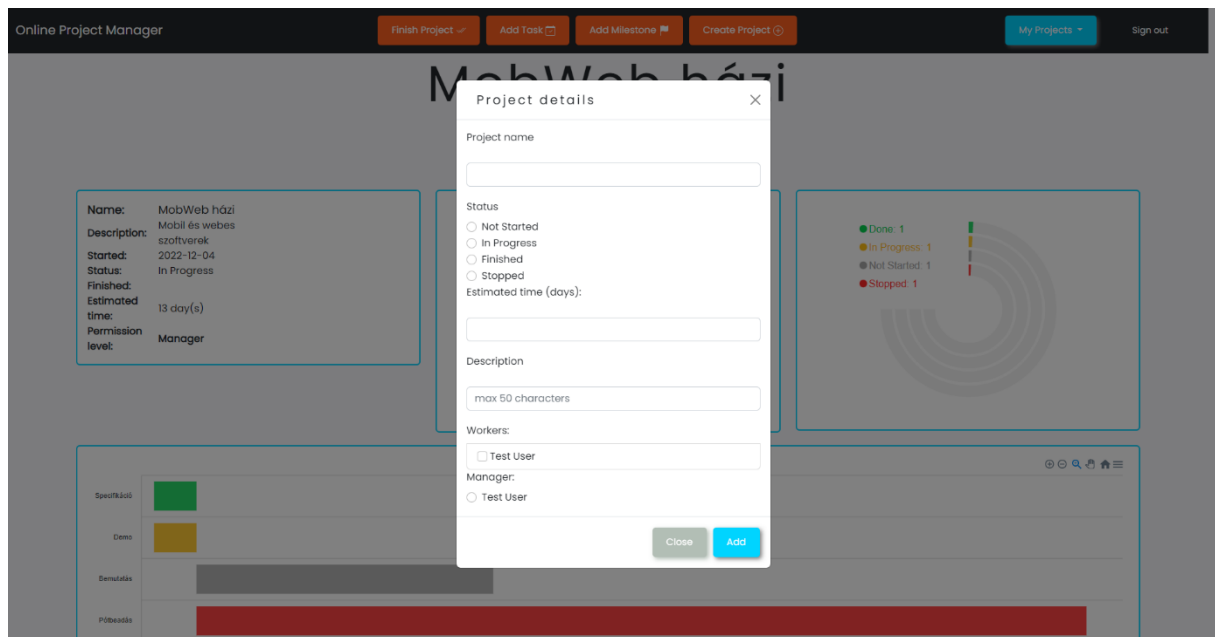


3. Az alkalmazás fő nézete

Belépést követően megjelenik az alkalmazás főképernyője, amely tartalmazza az adott projekthez tartozó taskok, illetve a mérföldkövek táblázatát. A Project Details menüpont alatt különféle adatok és diagramok jelennek meg, melyekhez ApexCharts wrappert használtunk.

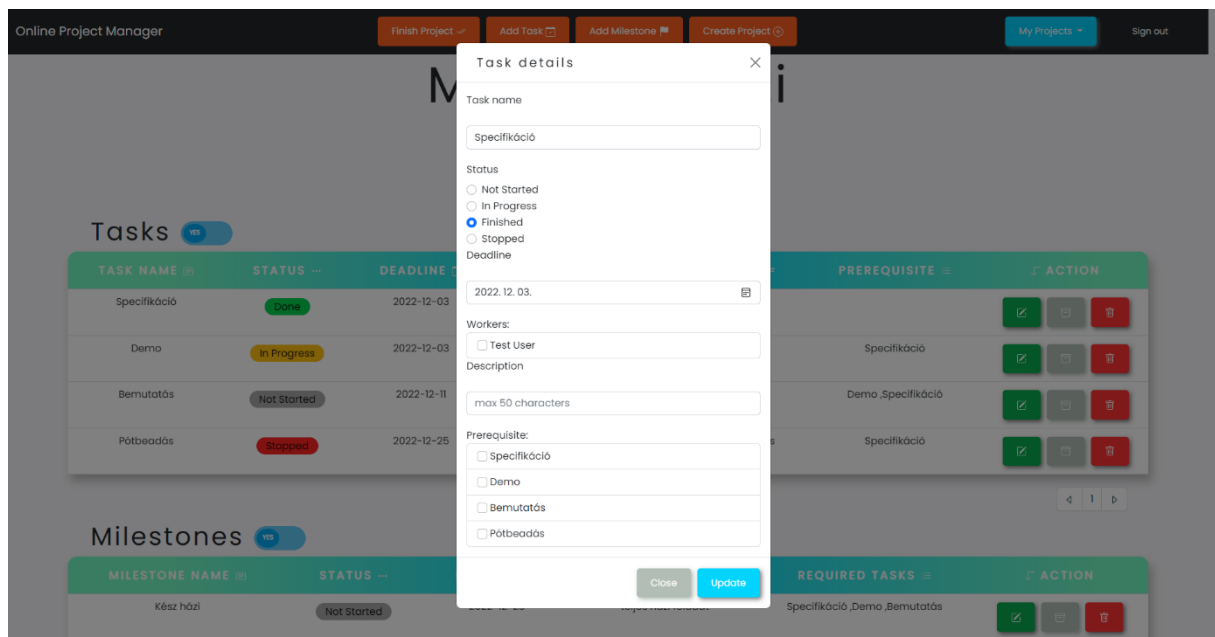


4. Project details nézet

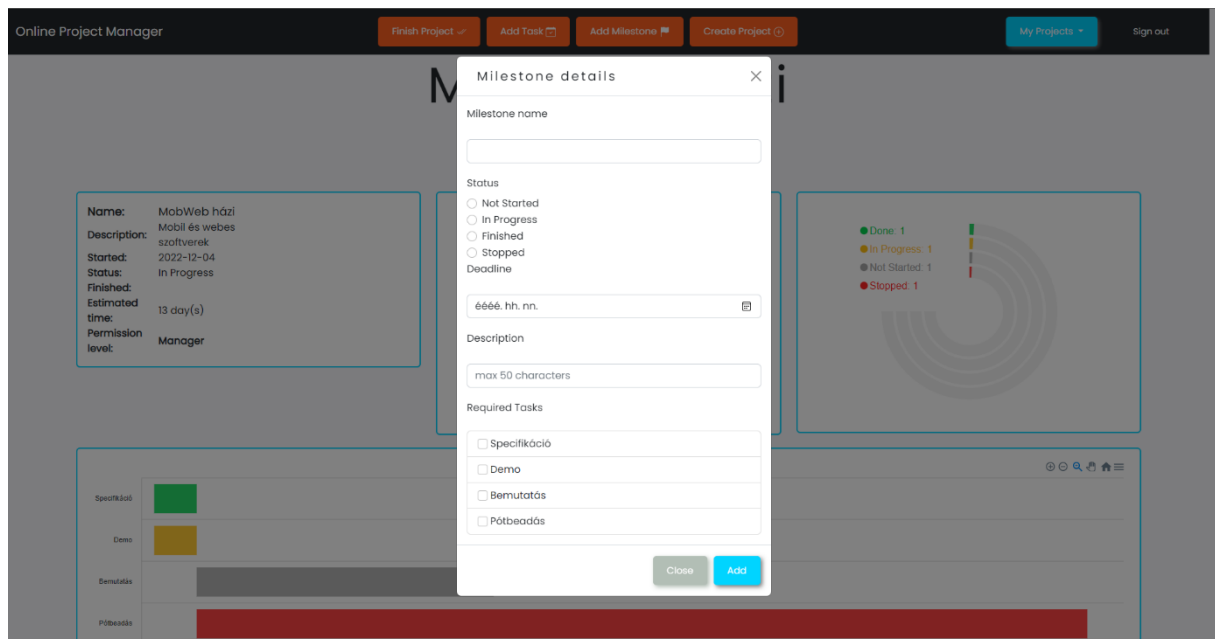


5. Projekt szerkesztése

lehetőség van továbbá projektek, taskok és milestoneok szerkesztésére, melyekhez egy-egy modal ablak jelenik meg. Ezekben az ablakokban lehetőség van többek között név, státusz, határidő, hozzárendelt dolgozók, előfeltételek hozzáadására/modosítására.



6. Task szerkesztése



1. Milestone szerkesztése

Sprint eltérítésként kapott feladat a taskok illetve milestoneok esetén jelzőlámpa jellegű mezők jeleznek az adott elem haladását. Ez a funkció már implementálva volt így teljesítette az elvárást.

Papós Zita:

Együtműködési platform:

Az együtműködési platform a Discord, illetve a Github volt. Mivel a munkánk nagyrésztét külön csináltuk, ezért ennél több nem kellett. A másokat is említő változásokról mindig értesítettük egymást, illetve időszakosan megnéztük egymás munkáját, annak fejlődését. Ha segítségre (más munkájának az értelmezésére) volt szükség, akkor hanghívásban azt hatékonyan adott időben megoldottuk. Nekem ez a fajta együtműködés tetszett, funkcionálisnak gondolom.

Csapatmunka, projekt:

A projekt nekem tetszett, láttam az értelmét. Egy hasznos alkalmazásnak tűnik mindennapos használatra is akár. Érdekes volt megismerni új fajta módszereket, pl. a Material-t, a Cookiek kezelését mobilon, az MVVM architektúrát.

A csapatmunka is jó volt, szerintem segítettünk egymásnak, amikor kellett, de nem pazaroltuk egymás idejét, ha nem.

Bui Quanganh Krisztián:

Együtműködési platform:

Gyors kommunikálásra Facebook Messenger-t használtunk, ha kisebb információkat kellett megosztani, megbeszélésekhez pedig Discord-ot használtunk, ahol egymásnak tudtunk segíteni. Az egész projektet GitHub-ra töltöttük fel, ahol elég könnyen tudtuk merge-olni a branch-eket, mivel mindenki kb. a saját "mappáján" (frontend, backend, Android alkalmazás) tudott dolgozni. Más együtműködési platformot nem használtunk, de lehet, hogy hasznos lett volna egy Kanban tábla, amivel könnyebben tudtuk volna jelezni egymásnak azt, hogy valamilyen feature kész van.

Csapatmunka, projekt:

Nekem a Node.js egy teljesen új dolog volt, így az első pár héten csak útmutatókat és videókat néztem, de nagyon motivált voltam közben. Megtanultam a REST API működését és felépítését, az autentikáció és autorizáció alapjait és még adatbázisokkal is kellett foglalkoznom. A csapat jó volt, mindenki rendesen csinálta a saját dolgát és ha valakinek problémája volt, akkor segítettünk egymásnak.

Fischer Balázs:

Együtműködési platform:

Az egyeztetés alapvetően messengeren folyt, majd a közös munkákhoz Discordot használtunk fel, a verziókezelést pedig Github segítségével csináltuk.

Csapatmunka, projekt:

Jól éreztem magam a projektben, gördülékenyen ment a csapatmunka, mindenki elérhető volt, amikor szükség volt egyeztetésre. Sokat tanultam a félév alatt, ami nagyrészt annak köszönhető, hogy lekötött a téma és foglalkoztatott is, hogy egy jó eredményt tudjak összehozni félév végére. Megtanultam az Angular keretrendszer alapvető működését, használatát, felépítését, ehhez kapcsolódóan pedig a TypeScript alapjait is, valamint a webdesignhoz szükséges Html és CSS

képességeim is szerintem elég jól fejlődtek. Használtam a program kivitelezéséhez ezek mellett még Bootstrapet, kipróbáltam a Material UI-t, különböző könyvtárakat, mint például az ikonok amik belekerültek a végső verzióba. A diagramok megvalósításához az ApexCharts wrappert használtam fel. Az utolsó demo után pedig kicsit belemélyedtem az Angular illetve CSS animációkba, ehhez elkezdtem az SCSS illetve SasS megtanulását is, ám ezek végül nem kerültek a programba. Mindennek az eredménye mostmár látható weboldalon: a Sign In oldalnak csináltam egy teljes redesignt amihez felhasználtam ezeket, módosítottam a UI-t hogy igényesebb, szebb és felhasználóbarátabb legyen. Majd ugyanezt megcsináltam a dashboard komponensre is, első betöltésnél animációk kerültek bele, valamint változások után is, javítottam a layouton hogy egységes, esztétikus legyen.

Imets Ákos:

Együttműködési platform:

A csapatmunka gond nélkül zajlott, csapaton belüli kommunikációhoz elsősorban Discordot valamint Messengert használtunk. Összességében ez is rendben volt, azonban előfordult párszor, hogy egy részfeladat, amin éppen dolgoztam már azelőtt belekerült a projektbe, hogy végeztem volna vele, és erről később értesültem.

Csapatmunka, projekt:

A témalabor felépítése és lezajlása nekem nagyon tetszett, hasznos volt, hogy hétről-hétre kaptuk a kitűzött feladatokat és visszajelzést az eddigi munkákról. Az Angular-rel és a frontend világával már korábban is meg akartam ismerkedni, úgyhogy ez egy ideális választás volt számomra.

Az Angular keretrendszer, illetve a TypeScript nyelv alapjait a félév elején sikerült elsajátítani az online segédanyagok segítségével, ezeket demo alkalmazásokon próbáltam ki. A frontend kinézetéhez először Angular Material UI -t használtam, azonban végül a Bootstrap mellett döntöttünk, így ez nem került bele a projektbe. A Bootstrap használatának következtében bővítettem a tudásom HTML-ből illetve részletesebben megismerkedtem a CSS-sel. A diagramok megvalósításához több verziót is kipróbáltam, például az npm gantt könyvtárat, vagy a DlhSoft gantt könyvtárat, végül azonban az ApexCharts került a végleges verzióba.

Projekt

Papós Zita	25%
Bui Quang Anh Krisztián	25%
Fischer Balázs	25%
Imets Ákos István	25%

Frontend

Fischer Balázs	55%
Imets Ákos István	45%