

**SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM
MAROSVÁSÁRHELYI KAR,
INFORMATIKA SZAK**



SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM

Flexibilis munkaprogram követés

DIPLOMADOLGOZAT

Témavezető:
Osztián Erika,
Egyetemi adjunktus

Végzős hallgató:
Kocsis Loránd

2023

UNIVERSITATEA SAPIENTIA DIN CLUJ-NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE,
SPECIALIZAREA INFORMATICĂ



UNIVERSITATEA
SAPIENTIA

Urmărirea unui program flexibil

LUCRARE DE DIPLOMĂ

Coordonator științific:
Osztián Erika,
Lector universitar

Absolvent:
Kocsis Loránd

2023

**SAPIENTIA HUNGARIAN UNIVERSITY OF
TRANSYLVANIA
FACULTY OF TECHNICAL AND HUMAN SCIENCES
COMPUTER SCIENCE SPECIALIZATION**



SAPIENTIA
HUNGARIAN UNIVERSITY
OF TRANSYLVANIA

Flexible work program follower

BACHELOR THESIS

Scientific advisor:
Osztián Erika,
Lecturer

Student:
Kocsis Loránd

2023

LUCRARE DE DIPLOMĂ

Coordonator științific:
Dr. Osztián Erika

Candidat: Kocsis Loránd
Anul absolvirii: 2023

a) Tema lucrării de licență:

Urmărirea unui program flexibil

b) Problemele principale tratate:

- Planificarea și organizarea programului flexibil
- Gestionarea programelor de lucru flexibile
- Tehnologia și infrastructura necesare

c) Desene obligatorii:

- Arhitectura ERP
- Structura bazei de date

d) Softuri obligatorii:

Modul bazat pe Odoo 7.0 care rezolvă problemele unui program flexibil, urmărirea intrărilor și ieșirilor personalului nu este o sarcină simplă mai ales atunci când locul de muncă permite munca la domiciliu. Acesta oferă departamentului de contabilitate posibilitatea de a crea un pontaj pentru perioade sau în intervale specificate.

e) Bibliografia recomandată:

<https://www.netsuite.com/portal/resource/articles/erp/what-is-erp.shtml>

f) Termene obligatorii de consultații: 2-3 ori pe lună

g) Locul și durata practicii: Universitatea „Sapientia” din Cluj-Napoca,
Facultatea de Științe Tehnice și Umaniste din Târgu Mureș, sala / laboratorul 414

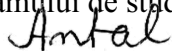
Primit tema la data de: 25.10.2022

Termen de predare: 15.06.2023

Semnătura Director Departament

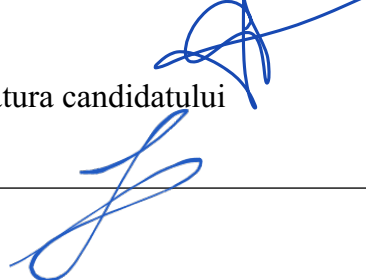


Semnătura responsabilului
programului de studiu



Semnătura coordonatorului

Semnătura candidatului



Declarație

Subsemnatul/a KOCSIS LORAND, absolvent(ă) al/a specializării
INFORMATICA, promoția 2023, cunoscând
prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a
Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta
lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală,
cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de
specialitate sunt citate în mod corespunzător.

Localitatea, TÂRGU MUREȘ
Data: 25.06.2023

Absolvent

Semnătura.....[Signature]

Kivonat

A dolgozat témája a flexibilis munkaprogram követése, amely megoldást nyújt azokra a problémákra, amikor egy munkahelyen rugalmas munkaidő van, és a dolgozók be- és kilépéseinek nyilvántartása nem egyszerű feladat, különösen akkor, ha a munkahely engedélyezi a távmunkát, ami egyre gyakoribbá válik, főleg az irodai munkák esetében.

Ennek a problémának a megoldására két fő indok miatt van szükség: a cég nyomon tudja követni a dolgozók napi munkaidejét, valamint eleget tesz az állam által havonta előírt jelenléti ív benyújtási kötelezettségének.

A projekt célja egy viszonylag egyszerű megoldás nyújtása a dolgozók munkaidő-beosztásának és szüneteinek nyilvántartására, valamint a törvényes követelményeknek való megfelelésre. A dolgozóknak lehetőségük van egyetlen gombnyomással be- és kilépni a rendszerből. Emellett a számvitel számára lehetőséget biztosít a jelenléti ív dokumentumok létrehozására periódusokra vagy meghatározott időszakokra.

Az ilyen vállalati problémák megoldására az ERP (Vállalatirányítási Rendszer) rendszerek szolgálnak, így a dolgozatban létrehoztam egy adatbázissal kommunikáló modult, valamint egy weboldalt, amely kapcsolatban áll ezzel a rendszerrel.

Rezumat

Tema lucrării este monitorizarea programului de lucru flexibil, care oferă o soluție pentru problemele întâmpinate atunci când o locație de muncă adoptă un program flexibil și urmărirea intrărilor și ieșirilor angajaților devine o sarcină complexă, în special atunci când se permite și munca de acasă, ceea ce devine tot mai frecvent, în special în cazul muncilor de birou. Există două motive principale pentru care este necesară rezolvarea acestei probleme: compania poate urmări numărul de ore lucrate de către angajați pe zi și poate satisface cerința lunară a statului privind prezentarea foii de prezență.

Scopul proiectului meu este să ofer o soluție relativ simplă pentru monitorizarea orelor lucrate și înregistrarea pauzelor angajaților, precum și pentru a satisface cerințele legale. Angajații pot înregistra intrarea și ieșirea din sistem printr-o singură apăsare de buton. De asemenea, se oferă posibilitatea creării documentelor de foaie de prezență pentru perioade sau intervale de timp prestabilite, în scopul contabilității. Astfel de probleme corporative sunt rezolvate de sistemele ERP (Enterprise Resource Planning), așadar am creat în lucrare un modul care comunică cu o bază de date și un site web care interacționează cu acest sistem.

Abstract

The topic of my thesis is monitoring flexible work schedules, which provides a solution to the challenges encountered when a workplace adopts a flexible work program and tracking employees' entries and exits becomes a complex task, especially when remote work is allowed, which is becoming increasingly common, particularly for office jobs. There are two main reasons why solving this problem is necessary: the company can track the number of hours worked by employees per day and comply with the monthly requirement of submitting attendance sheets to the state.

The aim of my project is to offer a relatively simple solution for monitoring employees' working hours and recording their breaks, as well as meeting legal requirements. Employees can register their entry and exit from the system with a single button press. Additionally, the system provides the capability to generate attendance sheet documents for predetermined periods or time intervals, for accounting purposes. Such corporate issues are addressed by Enterprise Resource Planning (ERP) systems, so I have developed a module in the paper that communicates with a database and a website that interacts with this system.

Tartalomjegyzék

1. Bevezető	11
2. Használt programok, technológiák bemutatása	12
2.1. Odoo 7.0	12
2.2. ERP rendszerek	12
2.2.1. Az ERP rendszerek használatának előnyei:	13
3. Tervezés, megvalósítás és kivitelezés	15
3.1. A projekt célja	15
3.2. Elvárások	15
3.3. A UI megtervezése	15
3.3.1. Egyéni widgetek (Custom Widgets)	17
3.3.2. XML form, tree, search, calendar nézetek	18
3.4. Felhasználók létrehozása, csoportokra osztása	19
3.5. A reportok létrehozása	21
3.6. Fordítások	22
4. Adatbázis és adattárolás	28
4.1. PostgreSQL	28
4.2. Táblák és relációk struktúrája, létrehozása	29
5. Modul	32
5.1. Mi a modul	32
5.2. Az Odoo modulok használata és pozitívításai	33
6. Weboldal	35
6.1. Mi az XML-RPC és használata	35
6.2. Weboldal UI és szükséglete	36
7. Továbbfejlesztési lehetőségek	38
7.1. Modul optimalizálás, kibővítés	38
7.2. Modullal kommunikáló applikációk	38
Összefoglaló	40
Köszönetnyilvánítás	41
Ábrák jegyzéke	42

1. fejezet

Bevezető

Minden cégnek az állam kötelezi, hogy minden hónapban mutasson fel minden munkásáról jelenléti ívet vagyis, hogy a munkások milyen órák között és mennyit dolgoztak le naponta. Erre a problémára vannak már megoldások, de egyik sem túl hatékony. Egy excel fájlal az a probléma áll fenn, hogy ha egy nap több be és kijelentkezést kéne a munkás megadjon annak a létrehozása nem lenne praktikus idő szempontjából. Ha egy be és kijelentkezést adunk meg az nem mindig lenne reális, monotonitás-hoz vezetne, ami észrevehető és megkérdőjelezhető, valamint a munkások nagy része nem mindig töltené ki.

Ennek a problémának a gyorsítására és hatékonyabb működése érdekében volt szükség egy rendszerre, ahol a munkások egy gombnyomásra be és ki tudnak jelentkezni. Ennek a problémának a megoldására létrehoztam egy modult, ahol a munkások be és kijelentkezési rekordokat tudnak felvezetni. Amelyek elmentődnek egy adatbázisba olyan adatokkal mint például mikor történt ki vagy bejelentkezés milyen dátumon, időponton, helyiségen, valamint mikor jött létre a rekord satöbbi. Eme pontos adatokkal könnyen követhető a munkások programja, valamint generálható a jelenléti ív az adott periódusra.

Ennek a megoldására az ERP rendszert használtam, amely az ilyen vállalati problémák megoldására hivatott. Felmerülhet bennünk a kérdés, hogy mi is az az ERP. A [vállalat irányítási rendszer](#) weboldala szerint:

„Az ERP (ERP system) egy angol betűszó, ami az Enterprise Resource Planning kifejezés rövidítése. Magyarul Vállalati Erőforrás Tervezés, de általában vállalatirányítási rendszernek szokták nevezni.

A dolgozat további részében bemutatom az említett rendszert, modult.

2. fejezet

Használt programok, technológiák bemutatása

2.1. Odoo 7.0

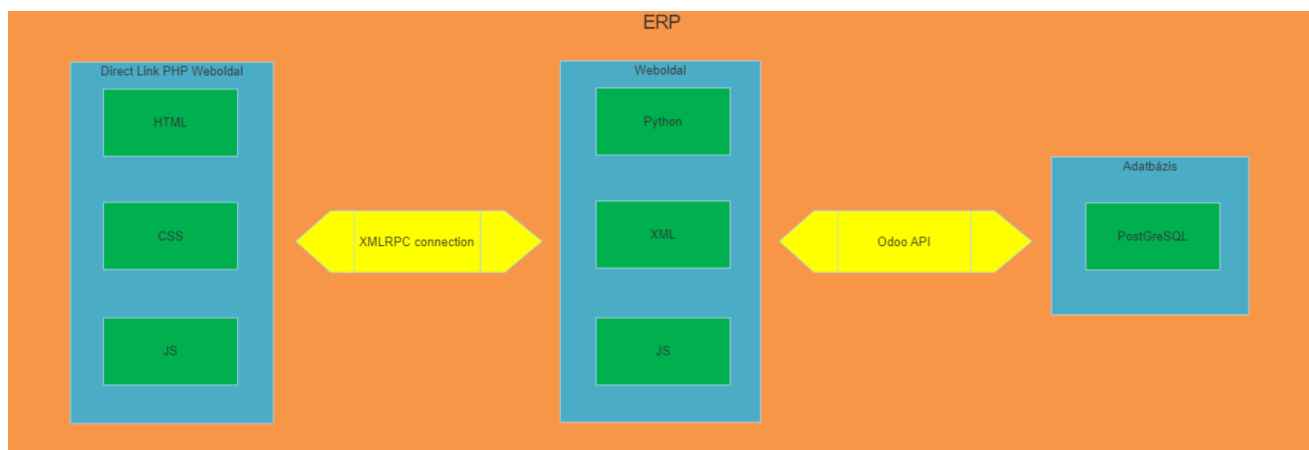
Dolgozatom ezen részében az általam használt Odoo 7.0 platformot mutatom be. Az Odoo [[Odo](#)] (korábbi nevén OpenERPként ismert) a kezdetektől fogva nyílt forráskódú szoftverként adta ki az alap szoftvert. Az Odoo ERP ([Enterprise Resource Planning](#)) rendszer egy vállalati erőforrás-tervezési szoftver, amelyet az üzleti folyamatok irányítására használnak az egész vállalatra kiterjedően. Az Odoo zökkenőmentesen képes integrálni funkcionális üzleti alkalmazásokat, úgynevezett Odoo appokat biztosít, amelyek kombinálva ERP-megoldást alkotnak. Az Odoo egy szerver alapú rendszer aminek elsődleges programozási nyelve a Python, valamint vizualizációra xml-t használ.[[Odob](#)]

2.2. ERP rendszerek

Az ERP rendszerek alapvetően olyan alkalmazások, amelyek automatizálják az üzleti folyamatokat, valamint betekintést és belső ellenőrzést biztosítanak, egy központi adatbázisra támaszkodva, amely összegyűjti az inputokat, beleértve a könyvelést, a gyártást, az ellátási lánc menedzsmentet, az értékesítést, a marketinget és az emberi erőforrásokat azok menedzselését és számontartását. Az ERP-rendszerek az erőforrások bölcs felhasználására törekvő vállalkozások számára mára alapvető fontosságúvá váltak. Segítségükkel a vezetők átcsoportosíthatják az emberi és pénzügyi tőkét, vagy hatékonyabb alapvető üzleti folyamatokat építhetnek ki, amelyek a minőség vagy a teljesítmény feláldozása nélkül pénzt takarítanak meg. Az ERP a tervezés és a koordináció terén is előnyös. Az alkalmazottak részletesen láthatják az aktuálisan rendelkezésre álló készletet és a vevői megrendeléseket, majd összehasonlíthatják a beszállítói megrendeléseket és az előre jelzett jövőbeli keresletet. Szükség esetén kiigazításokat végezhetnek a problémák elhárítása érdekében. Az ERP-szoftver javítja a kommunikációt és az együttműködést is, mivel a dolgozók ellenőrizhetik más részlegek helyzetét, hogy saját döntéseiket irányítani tudják.

2.2.1. Az ERP rendszerek használatának előnyei:

- **Költségmegtakarítás.** Az ERP-rendszerek talán legnagyobb értékét az adja, hogy számos módon pénzt takaríthat meg a szervezetnek. Számos egyszerű, ismétlődő feladat automatizálásával minimálisra csökkentheti a hibákat és azt, hogy az üzleti növekedéssel azonos ütemben kelljen új alkalmazottakat felvenni.
- **Jelentések/analitika.** Az adatok csak akkor hasznosak, ha a vállalatok elemezni és megérteni tudják őket, és egy ERP segít ebben. Valamint a könyvelőségi jelentéseket is létrehozhat ezzel megkönnyítve az adatok összegyűjtését és feldolgozását.
- **A munkafolyamatok átláthatósága.** Mivel minden munkafolyamat és információ egy helyen van, a rendszerhez hozzáféréssel rendelkező alkalmazottak láthatják a projektek állapotát és a különböző, a munkájuk szempontjából releváns üzleti funkciók teljesítményét.
- **Szabályozási megfelelés és adatbiztonság.** A pénzügyi beszámolási szabványok, valamint a kormányzati és iparág-specifikus adatbiztonsági előírások gyakran változnak, és egy ERP segíthet abban, hogy a vállalat biztonságban és megfelelésben maradjon. Az ERP ellenőrzési nyomvonalat biztosít azáltal, hogy nyomon követi az egyes tranzakciók életciklusát, beleértve az előírt jóváhagyási munkafolyamatok betartását.
- **Testreszabás, rugalmasság.** Az ERP rendszereknek a felépítése úgy van kialakítva, hogy viszonylagosan könnyen lehessen modifikációkat vagy új funkciókat hozzáadni.
- **Közös adatbázis.** Az ERP számos előnye a közös adatbázisból ered, amely lehetővé teszi a szervezetek számára, hogy számos részlegből származó információkat központosítsanak. A valós idejű adatoknak ez az egyetlen forrása kiküszöböli a különálló adatbázisok kézi összevonásának szükségességét
- **Automatizálás.** Az ERP-szoftverek másik alapvető jellemzője az olyan ismétlődő feladatok automatizálásának képessége, mint a számlázás, a rendelés feldolgozás és a jelentéstétel.
- Az ERP szoftverek legjelentősebb előnye, hogy egymástól különálló megoldások helyett egy egységes rendszerben kezeli a szervezet összes folyamatát és egy adatbázisban tárolja az adatokat.[\[ERP\]](#)



2.1. ábra. ERP architektúra

3. fejezet

Tervezés, megvalósítás és kivitelezés

3.1. A projekt célja

A projekt célja megvalósítani egy olyan univerzális Odoo modult amely feltelepíthető bármely ERP platformra és minimális módosítással alkalmazható a munkások munkaprogramjának, bejelentkezésének és kijelentkezésének a felvezetésére, felügyelésére. Valamint ezen adatok felhasználásával jelenléti ívek generálására az adott periódusokra. Ennek megvalósítása érdekében vannak a megjelenési naplók amik kapcsolatban állnak a személyekkel valamint a helységekkkel ezeken kívül vannak a hozzáférési jogosultságok valamint a periódusokra való felosztások a jelenléti ív rekordok létrehozásánál.

3.2. Elvárások

Ez a fejezet a weboldal, modul pontos követelményeinek leírását, elmagyarázását foglalja össze. Szó lesz a projekt felépítéséről felhasznált elemektől és azok működésének elmagyarázásáról, bemutatásáról.

3.3. A UI megtervezése

A projekt egy fő menüből három cím és azok almenüiből épül fel. A főcím az maga a modul neve ami flexibilis vagyis akármilyen Odoo szerverre feltelepíthető és a felső menüsorba hozzáadódik a legvégére. Ehhez tartozik három fő menü amelyek tartalmazzak almenüket ezek felosszák a modul funkcióit három részre ezek:

- Attendance ahol a be és kijelentkezésekkel kapcsolatos opciókat listázzuk: Attendance ahol a be és kijelentkezés egy gombnyomásos felvezetése található és az Attendance logs ahol ezen adatoknak felvezetett listája található részletesebb nézetben.
 - Attendance Logs ahol a be és kijelentkezés egy gombnyomásos felvezetése található, valamint mutassa a státuszt is vagyis hogy történt-e mai bejelentkezés vagy nem ha igen akkor mutassa az eltöltött időt.
 - Checkin Checkout ahol a be és kijelentkezések listáját nézhessük meg részletes kilistázás formájában, valamint az admin felhasználók itt láthatják az összes felvezetett idő bejegyzéseket és lehetőségük van szűrésre, csoportosításra is

ezzel könnyebbé téve a keresést, átláthatóságot. A user felhasználók csak a saját idő bejegyzéseit láthatják.

- Configurations ahol beállítsuk a felhasználók szükséges adatait és a modul modifikációit amik a következők:
 - Persons ahol a munkások adatait adjuk meg, tároljuk itt állítsuk be az alapértelmezett helyiséget ami meghatározza az egy gombnyomásra be és kijelentkezésnél az alapértelmezett helyiséget.
 - Locations ahol beállítsuk a helyiségeket ahonnan a munkás bejelentkezhet és kijelentkezhet ez fontos arésről hogy látható legyen otthonról dolgozott-e vagy az irodából.
 - Periods ami felosztódik két almenüre ahol létrehozuk a periódusokat ezt felosztva évi periódusokra valamint hónapi periódusokra. Ezek a periódus felosztások fontos szerepet játszanak a jelenléti ív jelentések kilistázásában. Ahol a létrehozott periódust kell megadnunk az adatok kilistázására, lekérésére, jelenléti ív generálására.
 - Wifi Points ez egy továbbfejlesztési lehetőség aminek egyelőre nincs fontosabb szerepe több információért lásd Továbbfejlesztési lehetőségek fejezetet.
- Reporting ahol a jelenléti ívet tudjuk kigenerálni, kilistázni, módosítani.
 - List Attendance Logs ahol a idő bejegyzéseket le tudjuk kérni bármilyen általunk megadott adatok által, vagyis kötelezően megadandó adatok a kezdeti dátum, végzeti dátum valamint opcionálisan megadandó szűrés személy, helység alapján. Ez azért fontos hogy bármikor bármilyen adatot le tudjunk kérni és eme adatok alapján kérdőívet kigenerálni anélkül, hogy erre létrehozzunk egy periódust.
 - List Attendance Sheet ahol az idő bejegyzéseket tudjuk lekérni a kiválasztott periódus alapján és a lekért adatokból jelenléti ívet kigenerálni.

Ezen menük kattintásakor hívódnak meg az akciók amelyek kezelik, meghívják az adott model tree, form, search... nézeteit. Minden almenünek van egy akciója ami felel a nézetek (views) meghívásáért, nagyon fontos hogy az akció létrehozása nélkül a menük nem lesznek megjelenítve, csak akkor jelennek meg ha egy akció hozzá lett kötve az adott menühöz. [3.1](#)

```
<!-- Main menu -->
<menuitem id="eoATTENDANCE_menu" name="eoATTENDANCE"
  web_icon="data/elastoffice.png"/>

<!-- Menu under Main Menu -->
<menuitem id="attendance_menu" name="Attendance" sequence="20" parent
  ="eoATTENDANCE_menu"/>

<!-- Menu Attendance Logs under Attendance menu -->
<menuitem name="Attendance Logs"
  id="menu_attendance_logs"
```

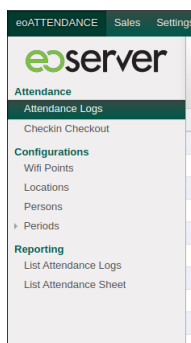


```

        action="attendance_logs_action_window"
        sequence="21"
        parent="attendance_menu"/>

<!-- Action of Attendance Logs -->
<record model="ir.actions.act_window" id="attendance_logs_action_window">
    <field name="name">Attendance Logs</field>
    <field name="res_model">attendance.logs</field>
    <field name="view_type">form</field>
    <field name="view_mode">tree,form,calendar</field>
</record>

```

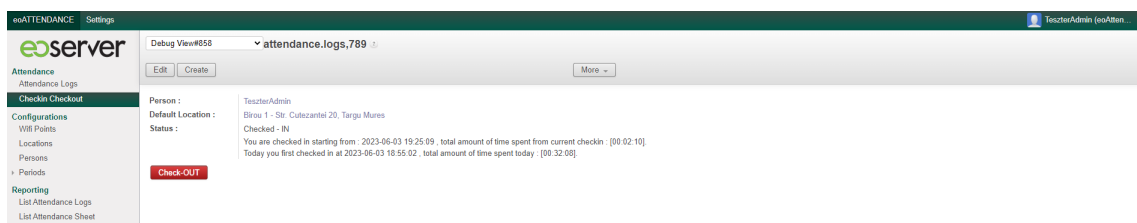


3.1. ábra. Menus

3.3.1. Egyéni widgetek (Custom Widgets)

A widget egyszerűen fogalmazva egy interaktív elem. A widget információkat jelenít meg, és elvárja, hogy a felhasználó valamilyen módon cselekedjen. Az egyéni widgetek olyan widgetek, ahol egy natív elemet a meghatározott specifikációtól eltérő módon használnak. Egy egyszerű példa erre egy gomb widget felhasználása linkként, ugyanis már nem ugyanazt a funkciót tölti be mint egy gomb.

Emme egyéni widgetre azért van szükség, hogy a bejelentkezés után tudjuk mutatni az idő múlását valós időben erre az Odoo fejlesztési felületen nincs létrehozva egy erre megfelelő widget ezért teljesen át kell alakítsunk egy meglévőt. Eme widget létrehozására html-t, css és legfőképpen JavaScript-et kell használnunk, html-t a nézet átírására, css-t a stílus megadására és JavaScript-et az eltelt idő valós idejű mutatásának leimplementálására.



3.2. ábra. Egyedi widget

3.3.2. XML form, tree, search, calendar nézetek

A form nézet határozza meg az adott rekord részletes nézetét valamint a rekord létrehozásánál, módosításánál is ez a nézet használandó fel kitöltendő mezőkként. (3.4)

```
<!-- Form view of Attendance Logs -->
<record model="ir.ui.view" id="attendance_logs_form_view">
  <field name="name">Attendance.logs.form</field>
  <field name="model">attendance.logs</field>
  <field name="arch" type="xml">
    <form string="Attendance Logs">
      <group colspan="1" col="1">
        <field name="image" widget="image" nolabel="1"/>
        <field name="persons_id" widget="selection" string="Person name"/>
        <field name="date" on_change="onchange_date(date)"/>
        <field name="date_type"/>
        <field name="loc_id" widget="selection" string="Location"/>
        <field name="category" string="Category"/>
      </group>
    </form>
  </field>
</record>
```

A tree nézet kiveszi az adott a modelből meghatározott elemeket és a rekordokat egy megadott sorrend alapján jeleníti meg a kiválasztott adatokkal. Ez a nézet hasznos keresések, szűrésekre egyszerűbb kinézetének hála könnyebben átlátható és ha az akcióhoz hozzá van csatolva egy form nézet akkor ha rákattintunk a tree nézetben lévő egyik rekordra átvezet annak a form nézetére. (3.7)

```
<!-- Tree view of Attendance Logs -->
<record model="ir.ui.view" id="attendance_logs_tree_view">
  <field name="name">Attendance.logs.tree</field>
  <field name="model">attendance.logs</field>
  <field name="arch" type="xml">
    <tree string="Attendance Logs">
      <field name="persons_id" string="Persons"/>
      <field name="date"/>
      <field name="date_type"/>
      <field name="loc_id" string="Location"/>
      <field name="category" string="Category"/>
    </tree>
  </field>
</record>
```

A search nézet határozza meg hogy milyen kereséseket, csoportosításokat tudunk végrehajtani az adott model keresőjében, fontos hogy minden oszlopot megadjunk ami alapján szeretnénk keresni, a filterezéssel tudunk szűréseket és csoportosításokat létrehozni. (3.5)

```
<!-- Search view of Attendance Logs -->
```

```

<record id="attendance_logs_search_view" model="ir.ui.view">
  <field name="name">Attendance.logs.search.form</field>
  <field name="model">attendance.logs</field>
  <field name="arch" type="xml">
    <search string="Attendance Logs">
      <field name="persons_id"/>
      <field name="loc_id"/>
      <filter string="Check-IN" name="date_type"
        domain="['date_type','=', 'Check - IN']" />
      <filter string="Check-OUT" name="date_type"
        domain="['date_type','=', 'Check - OUT']" />
      <group expand="0" string="Group By...">
        <filter string="Locations" domain="[]" context="{ 'group_by' :
          'loc_id' }" />
        <filter string="Type" domain="[]" context="{ 'group_by' :
          'date_type' }" />
        <filter string="Date/Time" domain="[]" context="{ 'group_by' :
          'date' }" />
        <filter string="Persons" domain="[]" context="{ 'group_by' :
          'persons_id' }" />
        <filter string="Category" domain="[]" context="{ 'group_by' :
          'category' }" />
      </group>
    </search>
  </field>
</record>

```

A calendar nézetben be tudjuk állítani hogy az adott model rekordjait behelyezze egy naptár nézetbe a könnyebben átláthatóság érdekében ahol tudunk szűrni is az adott kritériumok alapján. (3.6)

```

<!-- Calendar view of Attendance Logs -->
<record id="attendance_logs_calendar_view" model="ir.ui.view">
  <field name="name">Attendance.logs.calendar</field>
  <field name="model">attendance.logs</field>
  <field name="arch" type="xml">
    <calendar string="Attendance Calendar" color="persons_id"
      date_start="date" >
      <field name="persons_id"/>
      <field name="date_type"/>
    </calendar>
  </field>
</record>

```

3.4. Felhasználók létrehozása, csoportokra osztása

A csoportok felosszák a felhasználókat különböző jogosultságokkal rendelkező csoportokra ami meghatározza hogy milyen műveleteket képesek végrehajtani valamint milyen

rekordokat láthatnak. A csoportokra való felosztás fontos szerepet játszik a felhasználók jogosultságainak irányításában például egy konfigurációs rekordot a user felhasználóknak meg tudjuk adni hogy csak a sajátját lássa és tudja módosítani, az admin felhasználó azonban tudja látni az összeset és tudja módosítani is azokat. Ezen a példán a csoportok létrehozását láthatjuk valamint észrevehetjük, hogy az egyik csoport hivatkozik a másikra ez azért van mert az admin csoport örökli a user csoport jogosultságait és ezeket bővíti tovább :

```
<!--Users Groups-->
<record model="ir.module.category" id="module_category_eo_attendance">
  <field name="name">eoAttendance</field>
  <field name="sequence">7</field>
</record>

<record id="group_eoattendance_user" model="res.groups">
  <field name="name">User</field>
  <field name="category_id" ref="module_category_eo_attendance"/>
  <field name="implied_ids" eval="[(4, ref('base.group_user'))]" />
</record>

<record id="group_eoattendance_admin" model="res.groups">
  <field name="name">Admin</field>
  <field name="category_id" ref="module_category_eo_attendance"/>
  <field name="implied_ids" eval="[(4, ref('group_eoattendance_user'))]" />
  <field name="users" eval="[(4, ref('base.user_root'))]" />
</record>
```

A hozzáférési jogok (Access Rights) határozzák meg a létrehozott csoportok fennhatóságait az adott modulra ezek négy fontos restriktíót foglalnak magukba:

- Olvasási engedély ami meghatározza hogy az adott csoport láthassa-e a model rekordjait vagy nem.
- Írási engedély ami meghatározza hogy az adott csoport végrehajthat módosításokat létrehozott rekordokon.
- Létrehozási engedély ami meghatározza hogy az adott csoport létrehozhat új rekordokat.
- Törlési engedély ami meghatározza hogy az adott csoport kitörölhet meglévő rekordokat.

Ezek a restriktíók a megadott model összes rekordjára érvényesek.

```
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_attendance_logs_user,attendance.logs,
model_attendance_logs,group_eoattendance_user,1,1,1,1

access_attendance_logs_wizard_admin,attendance.logs.wizard,
model_attendance_logs_wizard,group_eoattendance_admin,1,1,1,1
```

A hozzáférési szabályok (Record rules) határozzák meg, hogy az adott csoportok milyen rekordokat láthatnak az adott modellen belül. Még az hozzáférési jogok (Access Rights) meghatározzák hogy milyen műveletek hajthatók végre addig a hozzáférési szabályok (Record rules) meghatározzák a modellben lévő rekordok láthatóságát.

Ezen a példán láthassuk hogy a user felhasználók csak az általuk létrehozott időbejegyzéseket láthassák az admin felhasználók pedig az összeset:

```
<!--Record rules-->
<record id="user_attendance_logs_record_rule" model="ir.rule">
  <field name="name">Acces of User to Attendance Logs</field>
  <field name="model_id" ref="model_attendance_logs" />
  <field name="domain_force">[( 'create_uid', '=', user.id)]</field>
  <field name="groups" eval="[(4,ref('group_eoattendance_user'))]" />
</record>

<record id="admin_attendance_logs_record_rule" model="ir.rule">
  <field name="name">Acces of Admin to Attendance Logs</field>
  <field name="model_id" ref="model_attendance_logs" />
  <field name="domain_force">[(1, '=', 1)]</field>
  <field name="groups" eval="[(4,ref('group_eoattendance_admin'))]" />
</record>
```

3.5. A reportok létrehozása

A reportok létrehozásánál átmeneti model (Transient Model) osztályokat használtam, amik arra jók hogy az ilyen modellekben lévő rekordok nem mentődnek le örökre. Az Odoo fejlesztési platform esetében például eme modellek úgy vannak lekezelve hogy egy ütemező (scheduler) ami egy megadott időközönként fut le és kitörli az adatokat. Ezeket a modelleket olyan esetekben szoktuk használni amikor az adott adatokra csak temporálisan van szükségünk mint például a jelentéseknél, adatok importálásánál vagy átmeneti lekérésénél.

Kétféle rekord nézet van a kettő között az adatok szűrésének a módszere ami megkülönbözteti őket egymástól. Az egyik ahol létrehozott periódusok által tudunk szűrni (3.12), a másik pedig ahol több lehetőségünk van a szűrésre (3.11). Erre a kettőre külön azért van szükség, hogy a könyvelőséghez kellő kigenerált jelenléti ív legyen elmentve periódus által, de a másik metódussal tudjunk periódustól független szűréseket csinálni, jelenléti ívet kigenerálni.

A jelenléti ív létrehozásánál a megadott szűrés alapján létrehozzuk a relációt a talált rekordokra és eme rekordokból állítjuk elő a jelenléti ívet (csv fájl formátumban) egy gombnyomásra. A csv fájl kigenerálása python-ban van megoldva függvénnyel amelyik hozzá van kötve egy xml rekordhoz, ami egy gombnyomás által hívódik meg:

```
<!-- Print Pontaj by Period -->
<record id="pontaj_report_xls" model="ir.actions.report.xml">
  <field name="report_type">xls</field>
  <field eval="[(6,0, [])]" name="groups_id"/>
  <field eval="0" name="multi"/>
</record>
```

```

    <field eval="0" name="auto"/>
    <field eval="1" name="header"/>
    <field name="model">list.attendance.sheet.reporting</field>
    <field name="name">Attendance sheet</field>
    <field name="report_name">pontaj_xls</field>
</record>

<!-- Print Pontaj by Custom Filter -->
<record id="attendance_logs_report_xls" model="ir.actions.report.xml">
    <field name="report_type">xls</field>
    <field eval="[(6,0,[])]" name="groups_id"/>
    <field eval="0" name="multi"/>
    <field eval="0" name="auto"/>
    <field eval="1" name="header"/>
    <field name="model">list.attendance.logs.reporting</field>
    <field name="name">Attendance Logs</field>
    <field name="report_name">eo_attendance_logs_report_xls</field>
</record>

```

3.6. Fordítások

Ezt a modult három nyelven lehet elérni Angol, Román és Magyar nyelvre van lefordítva. A fordításokra az Odoo-nak külön csomagja (openerp.tools.translate) van amivel meg tudjuk adni a fordítandó szövegeket amit összegyűjt egy po kiterjesztésű fájlba, ahol meg tudjuk adni az összegyűjtött szövegek adott nyelvre való fordításait.

Pelda po fájlra:

```

#: field:attendance.sheet.wizard,category:0
#, python-format
msgid "Category"
msgstr "Kategoria"

#. module: eo_attendance
#: model:ir.ui.menu,name:eo_attendance.menu_reporting_attendance_sheet
msgid "List Attendance Sheet"
msgstr "Jelenleti iv"

#. module: eo_attendance
#: model:ir.model,name:eo_attendance.model_attendance
#: model:ir.ui.menu,name:eo_attendance.attendance_menu
msgid "Attendance"
msgstr "Jelenlet"

#. module: eo_attendance
#: field:list.attendance.logs.reporting,date_from:0
msgid "From"
msgstr "Innentol"

```

3.3. ábra. Attendace Log Tree View

3.4. ábra. Attendace Log Form View

eoATTENDANCE

Sales

Settings

Admin

Administrator

eo

server

Debug View#637

Attendance Logs

1-80 of 152

Attendance

Attendance Logs

Checkin Checkout

Configurations

Wifi Points

Locations

Persons

Periods

Reporting

List Attendance Logs

List Attendance Sheet

Create

or Import

<input type="checkbox"/>	Persons	Date / Time	Type	Location	Category
<input type="checkbox"/>	Tesztler	03/27/2023 11:14:14	Check - OUT	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	03/27/2023 11:13:31	Check - IN	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	03/27/2023 11:12:14	Check - OUT	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	03/27/2023 11:10:22	Check - IN	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	TesztUser	01/26/2023 20:17:29	Check - OUT	Fabrica - Str. Ramurele 45, Targu Mures	Manual
<input type="checkbox"/>	TesztUser	01/26/2023 20:17:08	Check - IN	Fabrica - Str. Ramurele 45, Targu Mures	Manual
<input type="checkbox"/>	TesztlerAdmin	01/26/2023 20:15:47	Check - IN	Birou 1 - Str. Cutezantel 20, Targu Mures	Manual
<input type="checkbox"/>	TesztlerAdmin	01/26/2023 20:18:33	Check - OUT	Birou 1 - Str. Cutezantel 20, Targu Mures	Manual
<input type="checkbox"/>	TesztlerAdmin	01/26/2023 20:12:58	Check - IN	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	TesztUser	01/26/2023 20:07:59	Check - OUT	Fabrica - Str. Ramurele 45, Targu Mures	Manual
<input type="checkbox"/>	TesztUser	01/26/2023 20:07:46	Check - IN	Fabrica - Str. Ramurele 45, Targu Mures	Manual
<input type="checkbox"/>	TesztUser	01/26/2023 20:07:36	Check - OUT	Fabrica - Str. Ramurele 45, Targu Mures	Manual
<input type="checkbox"/>	TesztUser	01/26/2023 20:07:14	Check - IN	Fabrica - Str. Ramurele 45, Targu Mures	Manual
<input type="checkbox"/>	Tesztler	01/17/2023 15:43:33	Check - OUT	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	01/17/2023 08:35:28	Check - IN	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	01/16/2023 16:34:55	Check - OUT	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	01/16/2023 12:00:20	Check - IN	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	01/16/2023 12:00:15	Check - OUT	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	01/06/2023 16:05:25	Check - OUT	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	01/16/2023 09:57:03	Check - IN	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	01/06/2023 15:13:46	Check - IN	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	11/29/2022 16:22:18	Check - OUT	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	11/29/2022 12:27:08	Check - IN	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	11/29/2022 08:35:30	Check - OUT	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	11/29/2022 08:35:10	Check - IN	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	11/29/2022 08:34:32	Check - OUT	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	Tesztler	11/29/2022 08:32:46	Check - IN	Poli 2 tg. Mures	Manual
<input type="checkbox"/>	TesztUser	10/17/2022 10:22:58	Check - OUT	Fabrica - Str. Ramurele 45, Targu Mures	Manual
<input type="checkbox"/>	TesztUser	10/17/2022 10:22:50	Check - IN	Fabrica - Str. Ramurele 45, Targu Mures	Manual

3.7. ábra. Attendance Logs View

eoATTENDANCE

Sales

Settings

Administrator

eo

server

Debug View#641

Configurations Persons

1-8

Attendance

Attendance Logs

Checkin Checkout

Configurations

Wifi Points

Locations

Persons

Periods

Reporting

List Attendance Logs

List Attendance Sheet

Create

or Import

<input type="checkbox"/>	Name	UserName	User	Default Location
<input type="checkbox"/>	Janos	Proba	Proba	Poli 2 tg. Mures
<input type="checkbox"/>	Tesztler	Pelda	Administrator	Poli 2 tg. Mures
<input type="checkbox"/>	Benedek	Kecskes	Proba	Birou 1 - Str. Cutezantei 20, Targu Mures
<input type="checkbox"/>	Tamas	Karoly	Proba	Fabrica - Str. Ramurele 45, Targu Mures
<input type="checkbox"/>	Marton	admin	Proba	Fabrica - Str. Ramurele 45, Targu Mures
<input type="checkbox"/>	TesztUser	TesztlerUser01	TesztlerUser1	Fabrica - Str. Ramurele 45, Targu Mures
<input type="checkbox"/>	TesztlerAdmin	TesztlerAdm	TesztlerAdmin	Birou 1 - Str. Cutezantei 20, Targu Mures
<input type="checkbox"/>	Kacso Iren	kacso.iren	Kacso Iren	Poli 2 tg. Mures

3.8. ábra. Persons View

eoATTENDANCE

Sales

Settings

Administrator

eo

server

Debug View#639

Configurations Locations

1-3

Attendance

Attendance Logs

Checkin Checkout

Configurations

Wifi Points

Locations

Persons

Periods

Reporting

List Attendance Logs

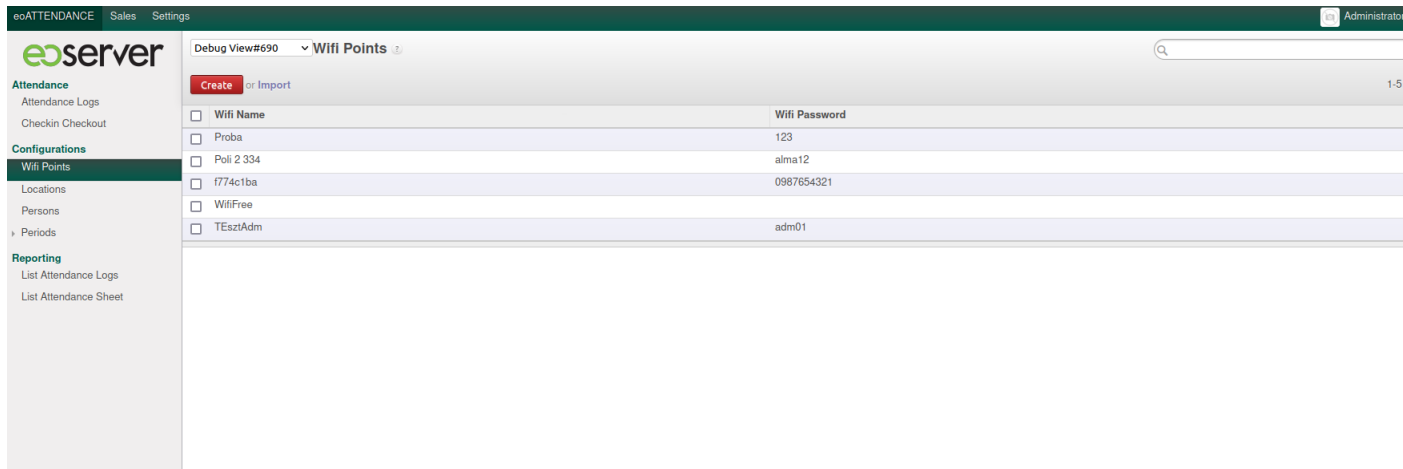
List Attendance Sheet

Create

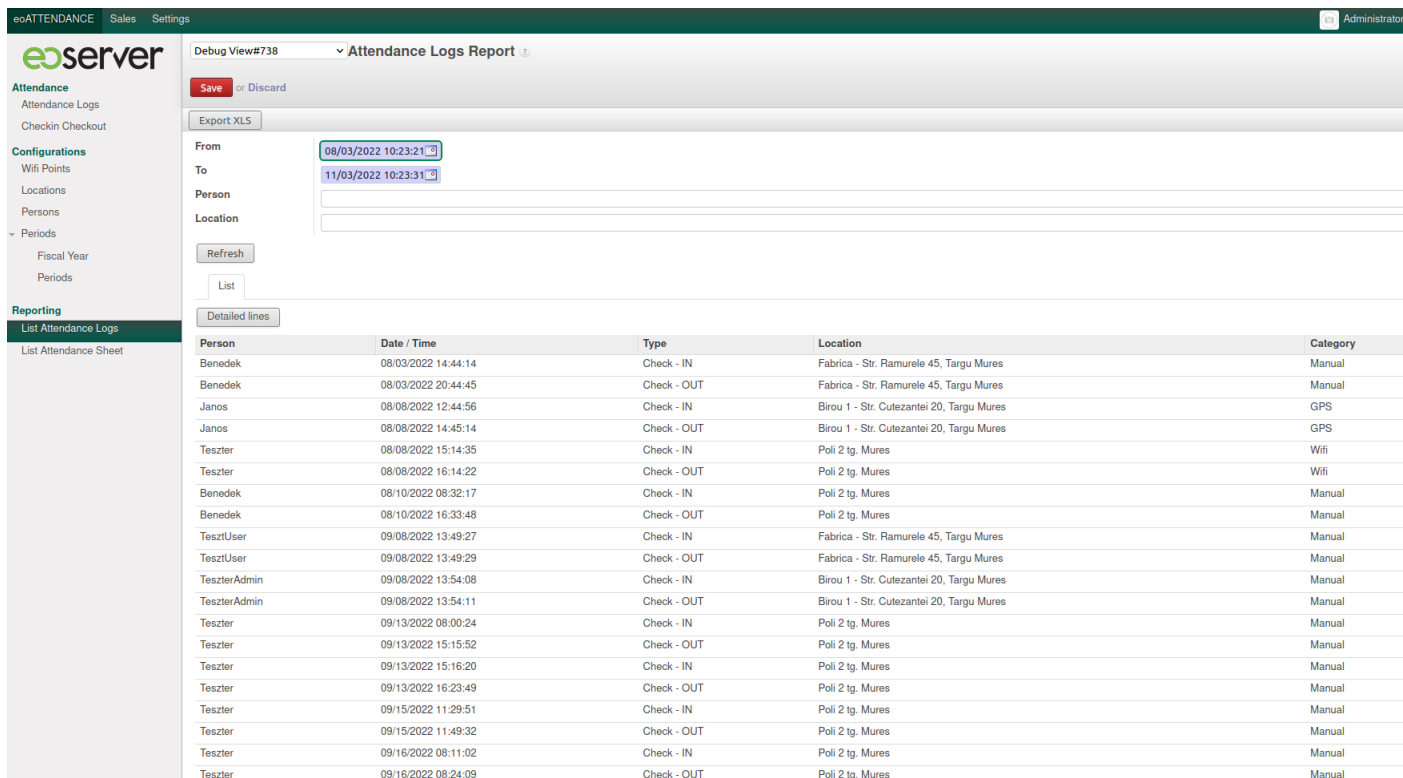
or Import

<input type="checkbox"/>	Name	Location Wifi Points	Description
<input type="checkbox"/>	Fabrica - Str. Ramurele 45, Targu Mures	1774c1ba	
<input type="checkbox"/>	Birou 1 - Str. Cutezantei 20, Targu Mures	WifiFree	
<input type="checkbox"/>	Poli 2 tg. Mures	Proba, Poli 2 334	Poli2

3.9. ábra. Locations View



3.10. ábra. Wifi Points View



3.11. ábra. Customized Report View

eoATTENDANCE

Sales

Settings

Administrator

eo

server

Attendance

Attendance Logs

Checkin Checkout

Configurations

Wifi Points

Locations

Persons

Periods

Fiscal Year

Periods

Reporting

List Attendance Logs

List Attendance Sheet

Debug View#747

Attendance Sheet Report

Save

or

Discard

Export XLS

Period

09/2022

Refresh

List

Detailed lines

Person	Date / Time	Type	Location	Category
TesztUser	09/08/2022 13:49:27	Check - IN	Fabrica - Str. Ramurele 45, Targu Mures	Manual
TesztUser	09/08/2022 13:49:29	Check - OUT	Fabrica - Str. Ramurele 45, Targu Mures	Manual
TesztlerAdmin	09/08/2022 13:54:08	Check - IN	Birou 1 - Str. Cutezantei 20, Targu Mures	Manual
TesztlerAdmin	09/08/2022 13:54:11	Check - OUT	Birou 1 - Str. Cutezantei 20, Targu Mures	Manual
Tesztler	09/13/2022 08:00:24	Check - IN	Poli 2 tg. Mures	Manual
Tesztler	09/13/2022 15:15:52	Check - OUT	Poli 2 tg. Mures	Manual
Tesztler	09/13/2022 15:16:20	Check - IN	Poli 2 tg. Mures	Manual
Tesztler	09/13/2022 16:23:49	Check - OUT	Poli 2 tg. Mures	Manual
Tesztler	09/15/2022 11:29:51	Check - IN	Poli 2 tg. Mures	Manual
Tesztler	09/15/2022 11:49:32	Check - OUT	Poli 2 tg. Mures	Manual
Tesztler	09/16/2022 08:11:02	Check - IN	Poli 2 tg. Mures	Manual
Tesztler	09/16/2022 08:24:09	Check - OUT	Poli 2 tg. Mures	Manual
Tesztler	09/16/2022 08:24:23	Check - IN	Poli 2 tg. Mures	Manual
Tesztler	09/16/2022 10:00:10	Check - OUT	Poli 2 tg. Mures	Manual
Tesztler	09/16/2022 10:35:05	Check - IN	Poli 2 tg. Mures	Manual
Tesztler	09/16/2022 12:55:49	Check - OUT	Poli 2 tg. Mures	Manual
Tesztler	09/19/2022 08:23:30	Check - IN	Poli 2 tg. Mures	Manual
Tesztler	09/19/2022 09:41:58	Check - OUT	Poli 2 tg. Mures	Manual
Tesztler	09/19/2022 09:42:31	Check - IN	Poli 2 tg. Mures	Manual
Tesztler	09/19/2022 11:45:28	Check - OUT	Poli 2 tg. Mures	Manual
Tesztler	09/19/2022 14:28:19	Check - IN	Poli 2 tg. Mures	Manual
Kacso Iren	09/19/2022 16:07:14	Check - IN	Poli 2 tg. Mures	Manual
Kacso Iren	09/19/2022 16:07:50	Check - OUT	Poli 2 tg. Mures	Manual

3.12. ábra. Report by Period View

4. fejezet

Adatbázis és adattárolás

4.1. PostgreSQL

Az adatbázis kulcsfontosságú, ugyanis ide tárolunk el minden adatot és innen is kérjük le. A PostgreSQL egy fejlett, vállalati szintű nyílt forráskódú relációs adatbázis, amely támogatja mind az SQL (relációs), mind a JSON (nem relációs) lekérdezést. Ez egy rendkívül stabil adatbázis-kezelő rendszer, amelyet több mint 20 évnyi közösségi fejlesztés támogat, ami hozzájárult a magas szintű rugalmasságához, integritásához és helyességéhez. A PostgreSQL-t számos webes, mobil, térinformatikai és analitikai alkalmazás elsődleges adattárolójaként vagy adattárházaként használják. [Posa]

A PostgreSQL használatának előnyei:

- A PostgreSQL gyors, a legtöbb kiadással a PostgreSQL egy új teljesítménynövelő funkcióval rendelkezik, amely jobb hatékonyságot tesz lehetővé a partíció kezelésével, a párhuzamos funkciókkal, az indexelés sebességével és az olyan párhuzamossági akadályok eltávolításával, mint például azok, amelyek miatt a tranzakcióknak várniuk kell egy művelet befejezésére.
- A PostgreSQL megbízható, a rendszerösszeomlások elleni védelem érdekében Write-Ahead Logot használ, így ha egy olyan tranzakciót kötöttek le, amelyet még nem írtak le az adatbázisba, a Write-Ahead Log újra lejátszódik, majd a tranzakciót lekötik.
- A PostgreSQL bővíthető, már számos beépített funkcióval rendelkezik, de nem korlátozódik ezekre. Bővítmények telepítésével új funkciókat adhat hozzá, hogy új funkciókat biztosítson. Ez a rendszer teszi lehetővé, hogy az olyan nagy projektek, mint a PostGIS, együtt tudjanak működni a PostgreSQL-el. A bővítmények új adattípusokat, operátorokat, függvényeket, index típusokat és még sok mást adhatnak hozzá.
- A PostgreSQL biztonságos, támogatja a SCRAM-SHA-256 hitelesítést, amely a régebbi és kevésbé biztonságos MD5 módszer helyett használható. De számos más lehetőség is rendelkezésre áll, például LDAP, Active Directory, RADIUS, tanúsítvány, csatlakoztatható hitelesítési modulok és számos más. Így a felhasználók kezelése a vállalat irányelveinek megfelelő módszerrel történhet.

- Nyílt forráskódú licenc. A PostgreSQL forráskódja nyílt forráskódú licenc alatt érhető el, ami biztosítja a szabad felhasználást, módosítást és implementálást.

[Posb]

4.2. Táblák és relációk struktúrája, létrehozása

Az Odoo(OpenERP) fejlesztői környezetben az adatbázisban lévő táblákat osztályok definiálásával hozzuk létre megadván nekik paraméternek a tábla típusát, például osv.osv egy permanens táblát készít ahol az adatok elmentődnek még az osv.memory egy átmeneti táblát hoz létre ami egy idő után törlődik. A tábla nevét a name változó definiálásával adjuk meg, az oszlopok nevét és típusát pedig a columns változóban adjuk meg a fields csomag használatával. Az order változóban állítsuk be, hogy az adatokat milyen sorrendben mutassa a vizuális felületen. Fontos tudni hogy az Odoo rendszer automatikusan hozzáadja minden táblához a következő oszlopokat: id ami a primary key, createDate, writeDate, createUid, writeUid.

```
from openerp.osv import fields, osv
from openerp.tools.translate import _
from openerp.tools import DEFAULT_SERVER_DATETIME_FORMAT

class attendance_logs(osv.osv):
    _name = "attendance.logs"
    _description = 'Attendance Logs'

    _columns = {
        'persons_id' : fields.many2one('configurations.persons',
            string='Person' , required=True ),
        'date' : fields.datetime('Date / Time' , required=True ),
        'date_type' : fields.selection([('Check - IN', 'Check - IN'), ('Check - OUT', 'Check - OUT')], string='Type' , required=True ),
        'image' : fields.binary('Attendance Image'),
        'loc_id' :
            fields.many2one('configurations.locations', string='Location' ,
                required=True),
        'category' : fields.selection([('GPS', 'GPS'), ('Manual', 'Manual'), ('Wifi', 'Wifi'), ('Other', 'Other')], string='Category' , required=True ),
        'status' : fields.selection([('Check - IN', 'Checked - IN'), ('Check - OUT', 'Checked - OUT')], string='Status'),
        'create_date': fields.datetime('Create Date', readonly=True),
    }

    _order = "create_date desc"
```

Az adott oszlopoknak sql constraint-eket is tudunk adni ahogy az látható az alábbi példán :

```
_sql_constraints = [
```

```
        ('username_key', 'UNIQUE (username)', 'You can not have two users with  
        the same username !')
```

```
]
```

A default változó segítségével be tudjuk állítani az alap értékét az adott oszlopoknak, erre használhatunk függvényeket is ahol ellenőrzések után visszatérítjük a megfelelő adatot. Ezek automatikusan meghívódnak amikor az Odoo create record funkciója meghívódik.

```
_defaults = {  
    'loc_id' : get_loc_id,  
    'date_type' : _is_check_in_out,  
    'persons_id' : _get_persons_id,  
    'date' : lambda *a: time.strftime('%Y-%m-%d %H:%M:%S'),  
    'category' : 'Manual',  
    'status' : is_check_in,  
}
```

Ahogy a fentebbi példákban is látható az Odoo fejlesztői rendszere megkönnyíti az adatbázis tábláinak azon oszlopainak, szabályainak a létrehozását és modifikálását. Az egyik nagy negativitás az adatbázis kezelő rendszerrel, az hogyha egy oszlopot hozzáadtunk a táblához és a kódból ezt kitöröljük attól még az adatbázisban még mindig jelen lesz még ki nem vesszük onnan sql parancs által.

5. fejezet

Modul

5.1. Mi a modul

Az Odoo alapvetően egy webalapú, nyílt forráskódú és moduláris alapú keretrendszer/szoftver, amely számos üzleti megoldást tartalmaz, mint például "CRM", "POS", "E-Commerce", "ERP", "Manufacturing", "Inventory Management", "Billing and Accounting" és "Project Management". Az Odoo-ban a modulok olyan önálló egységek, amelyek a funkciók és funkciók meghatározott csoportjait foglalják magukba.

Az Odoo modul egy olyan üzleti logika, amely segít a meglévő funkciók bővítésében vagy új funkciók hozzáadásában az Odoo-ban. Az Odoo modulok vagy teljesen új üzleti logikát adhatnak az Odoo rendszerhez, vagy módosíthatják és bővíthetik a meglévő üzleti logikát: létrehozható egy olyan modul, amely az adott országnak számviteli szabályait hozzáadja az Odoo általános számviteli támogatásához, míg a következő modul valós idejű megjelenítését támogatja. Ezek a modulok ha jól fel vannak építve akkor több Odoo szerveren is felhasználhatóak, a modulok azok amik megváltoztatják a szerver webes felületét vagy új dolgokat adnak hozzá. Az Odoo-ban tehát minden a modulokkal kezdődik és végződik. Minden modul egy adott üzleti folyamatra vagy alkalmazási területre összpontosít. Ezek a modulok telepíthetők, konfigurálhatók és integrálhatók, hogy az Odoo-t a vállalat egyedi követelményeihez igazítsák. [\[Modb\]](#)

Összefoglalva, az Odoo modulok a funkciók önálló egységei, amelyek hozzáadhatók, konfigurálhatók és testre szabhatók az Odoo ERP rendszeren belül. Moduláris és rugalmas megközelítést biztosítanak az Odoo képességeinek bevezetéséhez és bővítéséhez, hogy megfeleljenek a különböző iparágak vállalkozásainak egyedi igényeinek.

Háromféle modul létezik:

- 1) Alapmodulok: A Odoo beépített moduljait, amelyek az Odoo telepítési folyamatával együtt érkeznek, Alapmodulok(Core Modules) nevezzük.
- 2) Helyi modulok: A helyi modulokat mi magunk hozzuk létre helyileg. Ezeket a modulokat ugyanúgy beépítjük a programunkba, mint a beépített modulokat.
- 3) Harmadik féltől származó modulok: Az online elérhető és telepíthető modulokat harmadik féltől származó moduloknak nevezzük.

5.2. Az Odoo modulok használata és pozitívításai

Egy Odoo modul készítésére két fontos fájlra van szükségünk:

- 1) manifest.py: Ez a fájl segít megadni a modul metaadatait, és segít megjeleníteni a modult az Odoo alkalmazás listában. Az összes metaadat a szótárban van hozzárendelve.
- 2) init.py: Alapjában egy Python modulleíró fájl. Úgy működik, mint egy python modul, amely a program elejétől kezdve fut. Alapvetően az Odoo-ban betöltendő Python csomagok/fájlok importálásában segít.

Ez a két fájl szükséges ahhoz, hogy a modulok telepíthetők legyenek az Odoo-ban. [\[Moda\]](#)
Példa manifest fájlra:

```
{
    "name" : "eoAttendance",
    "version" : "1.0",
    "author" : "Elastoffice",
    "category" : "Extra Tools",
    "description": ""
}
eoAttendance
=====
Module for Attendance/Wifi/Location/ management. You can manage:

* Attendance logs
* Person
* Check In, Check out
* Location

"""
"depends" : ['base', 'crm', 'web'],
"init_xml" : [],
"js" : ["static/src/js/show_realtime.js",],
"qweb" : ["static/src/xml/*.xml",],
"demo_xml" : [],
"update_xml" : [
    "security/eoattendance_security.xml",
    "security/ir.model.access.csv",
    "menu.xml",
    "attendance_view.xml",
    "report/eoattendance_report.xml",
    "wizard/wizard_rp.xml",
    "show_messages.xml",
],
"installable": True,
'application': True,
'css' : ["static/src/css/show_realtime.css"],
}
```

Példa init fájlra:

```
import attendance
import report
import wizard
```

6. fejezet

Weboldal

6.1. Mi az XML-RPC és használata

Az XML-RPC egy olyan specifikáció és megvalósítások összessége, amely lehetővé teszi, hogy a különböző operációs rendszereken, különböző környezetekben futó szoftverek az interneten keresztül eljárást hívjanak.

Az XML-RPC lehetővé teszi a programok számára, hogy függvény vagy eljárás-hívásokat hajtsanak végre egy hálózaton keresztül. Az XML-RPC egy kis XML-szótárat használ a kérések és válaszok természetének leírására. Ez egy távoli eljárás-hívás, amely a HTTP-t használja szállításként és az XML-t kódolásként. Az XML-RPC-t úgy tervezték, hogy a lehető legegyszerűbb legyen, ugyanakkor lehetővé tegye az összetett adatszerkezetek továbbítását, feldolgozását és visszaküldését. [\[XML\]](#)

Ezt a weboldalt csak direkt linken keresztül lehet elérni valamint a direct link kell tartalmazza a felhasználó jelszavát és felhasználónevét, mert innen vesszük ki az XML-RPC által való bejelentkezéshez szükséges adatokat. Majd ezt eltároljuk egy session-ben és felhasználjuk a bejelentkezésnél. Az XML-RPC csatlakozik az Odoo szerverre és az ott lévő API függvényeket hívja meg. Vagyis a weboldal kommunikál az Odoo szerverrel ami kommunikál az adatbázissal.

```
// Check if username and password was given
if(isset($_GET["username"]) and isset($_GET["password"])){
    $_SESSION["login"] = $_GET["username"];
    $_SESSION["password"] = $_GET["password"];
    $login = $_SESSION['login'];
    $password = $_SESSION['password'];
}elseif(isset($_SESSION['login']) and isset($_SESSION['password'])){
    $login = $_SESSION['login'];
    $password = $_SESSION['password'];
}else{
    echo"Session expired or no username / password given !";
}

// login
$oe = new OpenERP($url, $db);
$oe->login($login, $password);
$context = array(
```

```

        'lang' => "en_US",
        'from_rpc' => 1
    );

    // Get user id
    $user_id = array(
        'model' => 'res.users',
        'fields' => array('id'),
        'domain' => array(array('login', '=', $login)),
    );
    $value_user_id = $oe->read($user_id);
    $user_id = $value_user_id['records'][0]['id'];

    // Get if user is admin or not
    $access_right = array(
        'model' => 'res.users',
        'fields' => array('groups_id'),
        'domain' => array(array('id', '=', $user_id)),
    );
    $access_right = $oe->read($access_right);

    // Call button functions
    if($_SERVER['REQUEST_METHOD'] == 'POST' and isset($_POST['checking_in'])) {
        checking_in($user_id, $oe, $context, $login, $password);
    }
    if($_SERVER['REQUEST_METHOD'] == 'POST' and isset($_POST['checking_out'])) {
        checking_out($user_id, $oe, $context, $login, $password);
    }
    if($_SERVER['REQUEST_METHOD'] == 'POST' and
        isset($_POST['export_attendance_sheet'])) {
        printing_attendance_sheet($user_id, $oe, $context, $periods);
    }
}

```

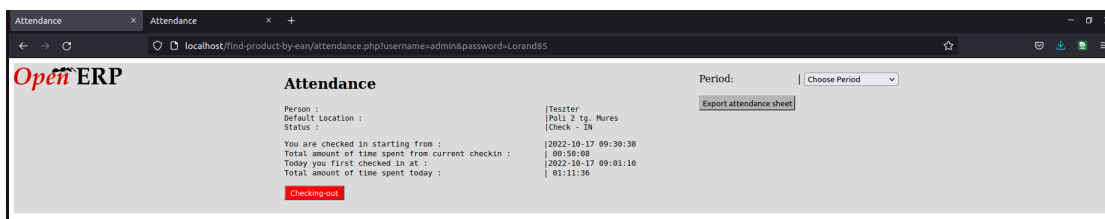
6.2. Weboldal UI és szükséglete

A weboldal lényege, hogy megkönnyítsük a felhasználók hozzáférését a be és kijelentkezés gombhoz. Az alapötlet az lenne, hogy direkt link által nyissa meg az oldalt vagy a böngésző megnyitásával automatikusan megnyílik és csak egy gombnyomást kell csináljon a be vagy kijelentkezéshez. Anélkül hogy külön beírja jelszavát és felhasználócímét ahogy azt az Odoo webes felületen kéri. Valamint nem kell attól félni hogy kijelentkeztetődik inaktivitásért, mert a weboldal minden kapcsolódásnál frissíti a csatlakozást az Odoo szerverhez.

A weboldal céges logó, szíkontrasztok flexibilisen kérés alapján változtatható mind a weboldalon és az Odoo szerver webes felületén is css használatával. Jelen pillanatban a kinézet szempontjából és funkcionalitásából megtartsa az Odoo weboldalas stílust.(6.1) Az admin felhasználóknak pedig külön szolgáltató gyors hozzáférést a jelenléti ív periódus szűrés által való kigenerálására.(6.2)



6.1. ábra. Weboldalak kinézete User felhasználónak



6.2. ábra. Weboldalak kinézete Admin felhasználónak

7. fejezet

Továbbfejlesztési lehetőségek

Ebben a fejezetben a module továbbfejlesztési lehetőségeiről illetve vele kommunikáló asztali applikáció létrehozásáról lesz szó. Ami az eddig elmagyarázott munkafolyamatot még jobban megkönnyítené és felgyorsítaná.

A modul továbbfejlesztésére több elgondolás is van maga a modul kibővítésétől kezdve más vele kommunikáló applikációk létrehozásáig:

7.1. Modul optimalizálás, kibővítés

- A weboldal direkt link által lehet elérni ennek jelen pillanatban van egy olyan hátránya, hogy a direkt link szabad szemmel is leolvasható, ezt a jövőre nézve minden képen meg kell oldani akár egy alap titkosítással is. Eddig nem volt nagy prioritás mert vagy otthonról, vagy az irodából lesz használva, ami esetben nem nagy rossz intencióval való leolvasás esélye, valamint a weboldalon nem lehet olyat csinálni, amit ne lehetne könnyen kijavítani.
- Szabadnapok felvezetésének lehetősége jelen pillanatban csak a be és kijelentkezéseket lehet felvezetni jó lenne ezt kibővíteni a szabadnapok felvezetésével betegszabadságok és a többi. Ez egy új modell osztályba lenne felvezetve ami kapcsolatban állna a személy osztállyal, kezdeti, végzeti dátuma, leírása oszlopai lennének.
- Reportok kibővítése, vizuális megjelenítések. Jelen pillanatban a reportoknál csak jelenléti ívet lehet kigenerálni azt is csak xls formátumban, ezt jó lenne kibővíteni pdf, csv formátumokkal, amiket egy felugró ablakban a print gomb megnyomásakor tudnánk kiválasztani. Valamint ha szükséges akkor személyekre osztva aktivitás vagy más diagramok kigenerálását is létrehozni.
- Jelen pillanatban a modul elérhető három fajta nyelven : Magyarul, Románul és Angolul. Ezeknek a kibővítése még több nyelvel, ha a kuncsaftok igényt tartanak rá.

7.2. Modullal kommunikáló applikációk

- Egy asztali applikáció ami XML-RPC által csatlakozik az Odoo szerverre akár csak a weboldal is teszi és a gép bekapcsolásakor elindul az applikáció és bejelentkezteti

a beállított felhasználóval. A háttérben futna és minden öt percben módosítaná az általa létrehozott kijelentkezés rekordnak dátumát. Ha több mint öt perc telt el akkor új be és kijelentkezést hoz létre és ezt ismételné. Ez a megoldás munkagépeken lenne a legpozitívabb ugyanis a felhasználó csak egyszer kell beállítsa az applikációt és az után egész végig működni fog, a be és kijelentkezések felvezetése kiküszöbölve az ki vagy bejelentkezés elfelejtését.

- Mobile applikáció ami XML-RPC által csatlakozna az Odoo szerverre akár csak a weboldal is teszi kétféle módon csinálna be és kijelentkezéseket vagy gps által való helység felismeréssel a beállított felhasználóra, vagy a wifi pontok alapján hozná létre. Viszont ez is akárcsak az asztali applikáció is egy időigényes és nagyobb továbbfejlesztés lenne viszont mindkettő felgyorsítaná és egyszerűbbé tenné a be és kijelentkezéseket. Egyszerűen mondva automatizálná őket, kiküszöbölné az elfelejtés problémáját.
- Ezzel a mobilos és asztali applikációval is az lenne a probléma, hogy mindkettőnek szükség lenne egy állandó wifi kapcsolatra. Vagy ennek a hibának fellépésének lekezelésére, adatok eltárolására amíg nem lesz wifi kapcsolat.

Összefoglaló

Dolgozatomban a jelenlét egyszerű felvezetésével és jelenléti ív kigenerálásával foglalkoztam, amelyet több különböző programozási technológia segítségével valósítottam meg. Elsőként bevezettem az általam használt technológiákat majd magyarázatot adtam ezeknek a működési elveire. Felvezetésre került az Odoo fejlesztési környezet, valamint az általa nyújtott szolgáltatások és ezek elkészítési módjai azok használatának magyarázata. Ezt követően elmagyaráztam az ERP rendszerek fogalmát, célját, használatát és előnyeit. Elmagyaráztam az általam készített projekt célját, tervezését, megvalósítását. Bemutattam a UI részeket, valamint elmagyaráztam és ábrázoltam őket. Ezek után elmagyaráztam az adatbázist erre felhasznált technológiákat, ábrázoltam a táblák felépítését. Ezt követően elmagyaráztam a modulok fontosságát és bemutattam a weboldalt és az ott felhasznált technológiákat.

A jövőre nézve szeretném bővíteni ennek a modulnak a funkcionalitásait, és szeretném elérni, hogy mint egy asztali applikáció lehessen használni.

A projekt modul és weboldal megtekinthetik egy GitHub repository-ban [ezen a linken](#).

Köszönetnyilvánítás

Külön köszönetet mondok az Elastoffice cégnek akik megadták a lehetőséget eme modul és weboldal megszületésére.

Ábrák jegyzéke

2.1. ERP architektúra	14
3.1. Menus	17
3.2. Egyedi widget	17
3.3. Attendace Log Tree View	23
3.4. Attendace Log Form View	23
3.5. Shearch Bar View	24
3.6. Calendar View	24
3.7. Attendance Logs View	25
3.8. Persons View	25
3.9. Locations View	25
3.10. Wifi Points View	26
3.11. Customized Report View	26
3.12. Report by Period View	27
4.1. Adatbázis struktúra (Részletes megtekintés)	31
6.1. Webolda kinézete User felhasználónak	37
6.2. Webolda kinézete Admin felhasználónak	37

Irodalomjegyzék

- [ERP] Erp rendszerek, <https://www.netsuite.com/portal/resource/articles/erp/what-is-erp.shtml>.
- [Moda] Creatig odoo module backend, <https://www.odoo.com/documentation/16.0/developer/tutorial>.
- [Modb] Creatig odoo module, <https://webkul.com/blog/how-to-create-a-module-in-odoo/>.
- [Odoa] Odoo főoldal, <https://www.odoo.com/>.
- [Odob] Odoo rendszerek, <https://hu.wikipedia.org/wiki/Odoo>.
- [Posa] Mi a postgresql, <https://aws.amazon.com/rds/postgresql/what-is-postgresql/>.
- [Posb] Postgresql, <https://www.enterprisedb.com/postgres-tutorials/why-more-and-more-enterprises-are-choosing-postgresql-their-go-database>.
- [XML] Mi az xml-rpc, <https://en.wikipedia.org/wiki/XML-RPC>.