

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ  
МЭДЭЭЛЛИЙН ТЕХНОЛОГИ, ЭЛЕКТРОНИКИЙН СУРГУУЛЬ  
МЭДЭЭЛЭЛ, КОМПЬЮТЕРЫН УХААНЫ ТЭНХИМ

Бат-Өлзий Лхагвадорж

**ERP хэрэглэгчийн портал**  
**(Self-Service Portal for ERP)**

Программ хангамж (D061302)  
Бакалаврын судалгааны ажил

Улаанбаатар

2025 оны 10 сар

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ  
МЭДЭЭЛЛИЙН ТЕХНОЛОГИ, ЭЛЕКТРОНИКИЙН СУРГУУЛЬ  
МЭДЭЭЛЭЛ, КОМПЬЮТЕРЫН УХААНЫ ТЭНХИМ

ERP хэрэглэгчийн портал  
(Self-Service Portal for ERP)

Программ хангамж (D061302)  
Бакалаврын судалгааны ажил

Удирдагч: \_\_\_\_\_ Б.Батням

Гүйцэтгэсэн: \_\_\_\_\_ Б.Лхагвадорж (22B1NUM7166)

Улаанбаатар

2025 оны 10 сар

# Зохиогчийн баталгаа

Миний бие Бат-Өлзий Лхагвадорж ”ERP хэрэглэгчийн портал” сэдэвтэй судалгааны ажлыг гүйцэтгэсэн болохыг зарлаж дараах зүйлсийг баталж байна:

- Ажил нь бүхэлдээ эсвэл ихэнхдээ Монгол Улсын Их Сургуулийн зэрэг горилохоор дэвшүүлсэн болно.
- Энэ ажлын аль нэг хэсгийг эсвэл бүхлээр нь ямар нэг их, дээд сургуулийн зэрэг горилохоор оруулж байгаагүй.
- Бусдын хийсэн ажлаас хуулбарлаагүй, ашигласан бол ишлэл, зүүлт хийсэн.
- Ажлыг би өөрөө (хамтарч) хийсэн ба миний хийсэн ажил, үзүүлсэн дэмжлэгийг дипломын ажилд тодорхой тусгасан.
- Ажилд тусалсан бүх эх сурвалжид талархаж байна.

Гарын үсэг: \_\_\_\_\_

Огноо: \_\_\_\_\_

ГАРЧИГ	ГАРЧИГ
<b>УДИРТГАЛ .....</b>	<b>1</b>
0.1 Судалгааны үндэс ба асуудлын тодорхойлолт .....	2
0.2 Судалгааны зорилго ба зорилт .....	2
0.3 Судалгааны арга зүй, хүрээ ба хязгаарлалт .....	3
<b>1. СУДАЛГАА БА ОНОЛЫН ҮНДЭС .....</b>	<b>5</b>
1.1 ERP системийн тухай өрөнхий ойлголт .....	5
1.2 Odoo ERP системийн бүтэц ба архитектур .....	6
1.3 Portal ба Controller .....	9
1.4 Хандалтын эрхийн загвар(ir.model.access, ir.rule) .....	12
<b>2. ENTERPRISE СИСТЕМИЙН ЛИЦЕНЗ АШИГЛАЛТ .....</b>	<b>17</b>
2.1 Байгууллагын ERP хэрэглээний өнөөгийн байдал (case-based судалгаа) .....	17
2.2 Лицензийн хүрээг багасгах .....	20
<b>3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР .....</b>	<b>22</b>
3.1 Системийн шаардлага .....	22
3.2 Ажлын явц .....	28
3.3 Системийн архитектурын өрөнхий бүтэц .....	34
3.4 Системийн архитектурын нарийвчилсан бүтэц .....	36
<b>ЭХ СУРВАЛЖУУД (АРА хэлбэрээр) .....</b>	<b>41</b>
<b>ХАВСРАЛТ .....</b>	<b>42</b>
A. СУДАЛГААНЫ ХҮЧНЭГТ, ЛИЦЕНЗИЙН ХАРЬЦУУЛАЛТ .....	43
B. OODOO ACCESS RULE ЖИШЭЭ .....	44
C. PORTAL CONTROLLER-ИЙН КОДЫН ХЭСЭГ (ЖИШЭЭ ХЭЛБЭРЭЭР) .....	45

ЗУРГИЙН ЖАГСААЛТ	ЗУРГИЙН ЖАГСААЛТ
1.1 Odoo ERP модулийн бүтэц . . . . .	8
3.1 MW Portal системийн ерөнхий архитектур . . . . .	35
3.2 Чөлөөний хүсэлт үүсгэх урсгалын дарааллын диаграмм . . . . .	37
3.3 Худалдан авалтын хүсэлт үүсгэх урсгалын дарааллын диаграмм . . . . .	38
3.4 Бараа материалын зарлага үүсгэх урсгалын дарааллын диаграмм . . . . .	38
3.5 Өгөгдлийн урсгал ба шалгалтын процесийн диаграмм . . . . .	39
3.6 Алдааны боловсруулалтын диаграмм . . . . .	40
C.1 Портал хэрэглэгчийн use case — хүсэлт илгээх ба дотоод урсгалтай уялдах жишээ . . . . .	45

## ХҮСНЭГТИЙН ЖАГСААЛ

3.1	MW Portal системийн функциональ шаардлагууд .....	23
3.2	MW Portal системийн функциональ шаардлагуудын нэгтгэсэн хүснэгт .	27
3.8	Portal хэрэглэгчийн түгээмэл урсгалуудын API холболт ба Odoo модель	36

# Кодын жагсаалт

1.1	Portal Controller-ийн жишээ . . . . .	10
1.2	Next.js талын API хүсэлтийн жишээ . . . . .	11
1.3	ir.model.access.csv жишээ . . . . .	14
1.4	ir.rule XML жишээ . . . . .	14
1.5	Практик Controller кодын жишээ . . . . .	15

## УДИРТГАЛ

Орчин үеийн байгууллагууд бизнесийн үйл ажиллагаагаа илүү үр ашигтай удирдах, мэдээллийг төвлөрүүлж боловсруулах, нөөцийн хуваарилалтыг оновчтой болгохи тулд Enterprise Resource Planning (ERP) системийг өргөнөөр ашиглаж байна. ERP систем нь байгууллагын санхүү, нягтлан бodoх бүртгэл, худалдан авалт, борлуулалт, агуулах, хүний нөөц зэрэг үндсэн үйл ажиллагааг нэгтгэн, нэгдсэн мэдээллийн орчин бүрдүүлдэг чухал программ хангамж юм.

Тэдгээрийн дотроос Odoo ERP нь нээлттэй эхийн бүтэцтэй, уян хатан, өргөтгөх боломжтой, олон салбарт ашиглагддаг систем бөгөөд дэлхийн 175 гаруй орны 7 сая гаруй хэрэглэгч ашиглаж байна. Odoo нь хоёр үндсэн хувилбартай: Community (үнэгүй, нээлттэй эхийн) ба Enterprise (лицензтэй, нэмэлт боломж бүхий) хувилбарууд.

Сүүлийн жилүүдэд Монголын байгууллагуудын дунд Odoo ERP Enterprise хувилбарыг ашиглах хандлага нэмэгдэж байгаа боловч судалгаанаас харахад эдгээр байгууллагуудын ихэнх нь Enterprise хувилбарын бүрэн боломжуудыг ашиглаж чаддаггүй, системийн үндсэн буюу Community түвшний үйлдлүүдээр хязгаарлагддаг байна. Энэ нь байгууллагууд өндөр өртөгтэй лицензийн системийг ашиглаж байгаа атлаа үнэ цэнийн хувьд бүрэн өгөөжөө гаргаж чадахгүй, нөөцийн үр ашиг буурахад хүргэдэг гол асуудал юм.

Иймээс Odoo ERP системийг илүү оновчтой, үр ашигтай ашиглахын тулд Portal хэрэглэгчийн орчныг хөгжүүлэх, зориулалтын Controller ашиглан хэрэглэгч ба харилцагчийн оролцоог нэмэгдүүлэх шаардлага зүй ёсоор тавигдаж байна. Portal орчин нь байгууллагын дотоод болон гадаад хэрэглэгчдэд зориулсан хялбар интерфэйс бүхий, зөвшөөрөгдсөн үйлдэл хийх боломжтой орчин бөгөөд Enterprise лицензийн төвөгтэй модуль, интерфэйсийг ашиглахгүйгээр тодорхой процессуудыг хэрэгжүүлэх боломжийг бүрдүүлдэг.

Энэхүү дипломын ажлын хүрээнд Odoo ERP Enterprise системийг ашиглаж буй байгууллагын зардлын оновчлол, хэрэглэгчийн туршлагыг сайжруулах зорилгоор Portal хэрэглэгчийн интерфэйс болон Controller-т суурилсан шийдэл боловсруулах болно. Судалгааны явцад Odoo системийн Community ба Enterprise хувилбарын ялгаа, зардлын бүтэц, хэрэглэгчийн хэрэгцээний түвшин,

мөн Portal хэрэглэгчийн үр нөлөө, системийн үр ашигт үзүүлэх нөлөөг шинжилнэ.

## 0.1 Судалгааны үндэс ба асуудлын тодорхойлолт

Одоогийн үед байгууллагуудын мэдээллийн урсгал улам бүр нэмэгдэж, ERP систем нь байгууллагын үйл ажиллагааг нэгтгэн удирдах гол хэрэгсэл болж байна. Гэсэн хэдий ч эцсийн хэрэглэгчид ERP системийг шууд ашиглахад төвөгтэй, олон шат дамжлагатай байх нь түгээмэл.

Иймээс ERP системийн өгөгдөлтэй шууд харилцах, хэрэглэгчдэд зориулсан хөнгөн, ойлгомжтой интерфэйс бүхий портал хэрэгцээтэй болж байна. Энэхүү портал нь Odoo erp-n controller ашиглан ERP системтэй холбогдож, ажилтанд ажлын цагийн хүсэлт илгээх, ирц бүртгэх болон худалдан авалтын мэдээлэлтэй ажиллах боломжийг олгох бөгөөд ERP-ийн хүргээж, үр ашгийг нэмэгдүүлнэ.

Монголын байгууллагууд ERP системийг өргөнөөр ашиглаж байгаа боловч заримдаа enterprise буюу лицензтэй хувилбар хэрэглэдэг. Энэ тохиолдолд ERP системд бүртгэлтэй хэрэглэгч бүр тусгай лиценз шаардах бөгөөд байгууллагын бүх ажилтанд бүрэн хэмжээний ERP лиценз олгох нь өртөг өндөртэй байдаг.

Гэвч олон ажилтанд ERP-ийн бүх функц шаардлагагүй бөгөөд зөвхөн цаг бүртгэх, цагийн хүсэлт илгээх, худалдан авалтын захиалга харах зэрэг боломжууд л хэрэгтэй байдаг. Иймээс ERP системийн лицензийн зардлыг нэмэгдүүлэхгүйгээр, зөвхөн шаардлагатай функциуудыг ашиглах боломж олгох тусгай портал хөгжүүлэх нь зүйтэй юм.

## 0.2 Судалгааны зорилго ба зорилт

Байгууллагын Odoo ERP системийн Enterprise хувилбарыг илүү оновчтой, зардал багатай ашиглах зорилгоор Portal хэрэглэгчийн орчныг odoo erp-n controller ашиглан шийдэл боловсруулахад оршино.

1. Odoo ERP систем хэрхэн ажилладаг түүний боломжуудыг судлах.

2. Байгууллагын Odoo ERP ашиглалтын бүтэц, хүндрэлүүдийг судлах.
3. Хэрэглэгч, харилцагчийн portal оролцоог нэмэгдүүлэх зориулалтын API болон интерфэйс бүтээх.
4. Боловсруулсан шийдлийн үр нөлөөг туршиж, системийн үр ашгийн үнэлгээ хийх.

### **0.3 Судалгааны арга зүй, хүрээ ба хязгаарлалт**

#### **Судалгааны арга зүй**

Энэхүү судалгаанд чанарын (qualitative) болон тоон (quantitative) шинжилгээний аргуудыг хослуулан ашиглав. Судалгааны үндсэн арга зүй нь баримт бичгийн судалгаа, системийн архитектурын шинжилгээ, болон бодит хэрэглээний кейс судалгаа (case study) дээр суурилсан болно.

Баримт бичгийн судалгааны хүрээнд Odoo ERP Enterprise лицензийн бүтэц, хэрэглэгчийн эрхийн загвар, Portal болон Controller-ийн үүргийг тодорхойлсон Odoo-ийн албан ёсны баримт бичиг, хөгжүүлэгчийн гарын авлага, мөн бусад ERP системийн лицензийн бодлого (SAP, Oracle, Microsoft Dynamics)-той харьцуулсан судалгааны материалуудыг ашигласан ([1]).

Системийн архитектурын шинжилгээний хүрээнд Odoo ERP 18 хувилбар дээр суурилсан enterprise орчны хандалтын эрхийн удирдлага (ir.model.access, ir.rule) болон Portal механизмын бүтэц-ийг техникийн түвшинд задлан шинжилж, хэрэглэгчийн оролцоог сайжруулах controller шийдлийг боловсруулсан.

#### **Судалгааны хүрээ**

Судалгааны ажлын хүрээ нь Odoo ERP Enterprise 18 хувилбар бөгөөд тухайн хувилбарын архитектур, лицензийн зарчим, хэрэглэгчийн эрхийн бүтэц, болон portal controller-ийн ажиллагаанд төвлөрсөн. Судалгаанд Монголын жижиг, дунд хэмжээний байгууллагуудыг төлөөлсөн кейс хэлбэрийн жишээг ашигласан бөгөөд эдгээр байгууллагуудын Odoo ERP Enterprise системийн ашиглалт, лицензийн бодит хэрэгцээний зөрүүг судалсан.

#### **Судалгааны хязгаарлалт**

Энэхүү судалгаа нь дараах хязгаарлалтуудтай:

- Судалгаа нь зөвхөн Odoo ERP Enterprise 18 хувилбарын архитектур болон лицензийн бүтэц дээр төвлөрсөн бөгөөд бусад ERP системийн техникийн гүйцэтгэлийн нарийвчилсан туршилт хийгдээгүй.
- Судалгааны зардлын тооцоолол нь байгууллагын өрөнхий статистик ба хэрэглэгчийн тоонд тулгуурласан бөгөөд бодит төслийн гүйцэтгэлийн нарийвчилсан тоон мэдээлэлд суурилаагүй.
- Portal controller-ийн шийдэл нь үйлдвэрлэлийн (production) орчинд бүрэн хэрэгжүүлэлт хийх.

# 1. СУДАЛГАА БА ОНОЛЫН ҮНДЭС

## 1.1 ERP системийн тухай ерөнхий ойлголт

Enterprise Resource Planning (ERP) систем нь байгууллагын нэгдсэн нөөцийн удирдлагын цогц шийдэл бөгөөд байгууллагын санхүү, нягтлан бодох бүртгэл, борлуулалт, худалдан авалт, агуулах, үйлдвэрлэл, хүний нөөц зэрэг үндсэн бизнесийн үйл ажиллагааг нэг мэдээллийн системд уялдуулан нэгтгэдэг программ хангамж юм. ERP системийн гол зорилго нь мэдээллийн давхардлыг арилгах, шийдвэр гаргалтыг хурдаасах, байгууллагын нөөцийн оновчтой хуваарилалт хийхэд оршино.

ERP системүүд нь ихэвчлэн модульчлагдсан бүтээцтэй байдаг бөгөөд байгууллага өөрийн хэрэгцээ, үйл ажиллагааны онцлогоос хамаарч санхүү, хүний нөөц, бараа материалын менежмент, борлуулалт, хэрэглэгчийн харилцааны менежмент (CRM) зэрэг модулиудыг сонгон ашигладаг. Модуль тус бүр нь байгууллагын дотоод процессын нэг хэсгийг удирдах бөгөөд эдгээр модулиуд хоорондоо нэг өгөгдлийн сантай, бодит цагийн уялдаатай байдлаар ажилладаг.

ERP системийн хамгийн чухал онцлог нь нэгдсэн өгөгдлийн сан юм. Өгөгдөл нэг л газарт хадгалагдаж, бүх модуль, хэлтэс, хэрэглэгчид ижил мэдээлэлд хандах боломжтой тул мэдээллийн зөрүү, давхардал, алдааны эрсдэл багасдаг. Энэ нь байгууллагын шийдвэр гаргалтын чанарыг сайжруулах, менежментийн түвшинд ил тод байдлыг нэмэгдүүлэхэд чухал ач холбогдолтой.

ERP системүүдийг арилжааны (proprietary) болон нээлттэй эхийн (open-source) гэсэн хоёр үндсэн төрөлд ангилдаг. Арилжааны ERP системд SAP ERP, Oracle ERP Cloud, Microsoft Dynamics 365 зэрэг томоохон төлөөллүүд ордог бөгөөд эдгээр нь лицензийн төлбөр, дэмжлэгийн зардал өндөр байдаг. Харин нээлттэй эхийн ERP системүүд болох Odoo, ERPNext, Dolibarr, Tryton зэрэг нь програмчлалын код нээлттэй, байгууллагын хэрэгцээнд тохируулан өөрчлөх боломжтой байдгаараа давуу талтай.

Сүүлийн жилүүдэд олон улсын байгууллагууд ERP системийг Software-as-a-Service (SaaS)

## 1. Судалгаа ба Онолын Үндэс

хэлбэрээр ашиглах хандлагатай болж байна. Энэ тохиолдолд байгууллага ERP системийг үүлэн (cloud) орчинд байршуулж, хэрэглэгч тус бүрийн лицензийн төлбөрөөр ашигладаг. Гэвч энэ хэлбэрийн сүл тал нь байгууллагын хэмжээнээс хамаарч лицензийн зардал өндөрсөх, мөн бүх хэрэглэгчдэд бүрэн эрх олгох шаардлага үүсдэг.

Ийм нөхцөлд байгууллагууд ERP системийн үр ашгийг нэмэгдүүлэхийн тулд хэрэглэгчийн хандалтын түвшинг оновчтой тогтоох, шаардлагатай тохиолдолд Portal буюу хөнгөн хэрэглэгчийн орчныг ашиглах замаар лицензийн зардлыг бууруулах стратегийг хэрэгжүүлж байна.

### 1.2 Odoo ERP системийн бүтэц ба архитектур

Odoo ERP нь байгууллагын бизнесийн процессуудыг автоматжуулах, нэгдсэн мэдээллийн урсгал бий болгох зорилготой модульчлагдсан, гурван шатлалт архитектуртай (Model–View–Controller) систем юм [1].

#### Model буюу бизнес логикийн түвшин

Model түвшин нь системийн гол цөм бөгөөд өгөгдлийн бүтэц, бизнесийн дүрэм, харилцан хамаарлыг тодорхойлдог. Odoo нь Python хэл дээр бичигдсэн ORM (Object Relational Mapping) системийг ашигладаг бөгөөд өгөгдлийн санд хийгдэх бүх үйлдэл (CREATE, READ, UPDATE, DELETE) нь ORM-оор дамждаг.

ORM нь PostgreSQL өгөгдлийн сантай нягт уялдаа бүхий тул өгөгдлийг SQL хэл бичихгүйгээр Python кодоор удирдах боломжийг олгодог. Жишээлбэл, `models.Model`-ийг өргөтгөн тодорхойлсон класс бүр нь өгөгдлийн сангийн нэг хүснэгттэй шууд харгалздаг.

#### Хүснэгтийн уялдаа холбоо (Relationships)

Odoo-ийн ORM нь өгөгдлийн хоорондын харилцан хамаарлыг гурван үндсэн төрлөөр тодорхойлдог бөгөөд эдгээр нь `many2one`, `one2many`, `many2many` талбаруудаар илэрхийлэгддэг.

- `many2one` — Нэг хүснэгтийн бичлэгийг нөгөө хүснэгтийн нэг бичлэгтэй холбох хамаарал.

Жишээлбэл, `hr.employee` модель нь ”`department_id = fields.Many2one('hr.department')`” гэж тодорхойлогдсон бол тухайн ажилтан нэг л хэлтэст харьялгадана.

## 1. Судалгаа ба Онолын Үндэс

- **one2many** — Нэг бичлэгийг олон бичлэгтэй холбох хамаарал. Жишээ нь, `hr.department` моделийн `employee_ids = fields.One2many('hr.employee', 'department_id')` гэж тодорхойлогдвол тухайн хэлтэст олон ажилтан хамаарна.
- **many2many** — Хоёр хүснэгтийн хооронд олон-олон харилцаа үүсгэх. Жишээлбэл, `res.users` ба `res.groups` хоорондын хамаарал: нэг хэрэглэгч олон group-д харьялгадж, нэг group олон хэрэглэгч агуулж болно.

Эдгээр хамаарлууд нь өгөгдлийн логик уялдаа, бүтцийн нэгтгэл, мэдээллийн дахин ашиглалтыг хангадаг бөгөөд Odoo ERP системийн модуль хоорондын холболтын үндэс болдог.

### **View буюу хэрэглэгчийн интерфэйсийн түвшин**

View түвшин нь хэрэглэгчийн системтэй харилцах интерфэйс бөгөөд XML хэл дээр тодорхойлогддог. Хэрэглэгчийн харагдац (жагсаалт, форм, канбан, календар, график) нь Odoo-ийн QWeb болон OWL (Odoo Web Library) фреймворкуудаар дамжин рэндэрлэгддэг.

View нь бизнесийн өгөгдлийг ойлгомжтой байдлаар үзүүлэхээс гадна role-based (эрхийн түвшин хамаарах) харагдацын ялгааг дэмждэг. Жишээлбэл, Portal хэрэглэгч зөвхөн өөрийн хүсэлтийн мэдээллийг, харин дотоод ажилтан бүх хэлтсийн өгөгдлийг харах эрхтэй байж болно.

### **Controller буюу удирдлагын түвшин**

Controller нь HTTP хүсэлт болон системийн өгөгдлийн түвшний логикийг холбодог дундын давхарга юм. `@http.route` декоратор ашиглан хүсэлтийн зам, зөвшөөрлийн түвшин (`auth='user'`, `auth='portal'`, `auth='public'`) болон хүсэлтийн төрлийг (GET, POST) тодорхойлдог.

Controller нь веб эсвэл API орчны (жишээ нь React, Next.js) хүсэлтийг хүлээн авч, ORM руу чиглүүлж, JSON эсвэл HTTP хариу буцаадаг. Энэ механизм нь Portal хэрэглэгчийн орчны үндэс болж, Enterprise лицензгүй хэрэглэгчдэд хязгаарлагдмал CRUD үйлдэл хийх боломжийг олгодог.

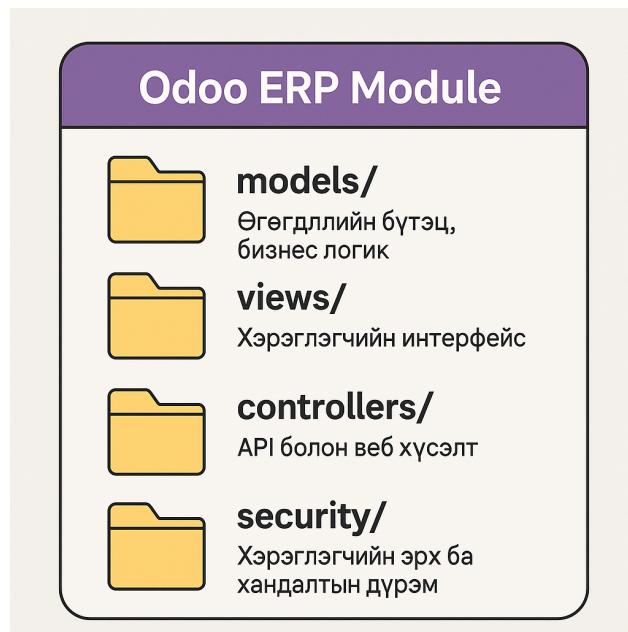
### **Модульчлагдсан бүтэц ба өргөтгөх боломж**

## 1. Судалгаа ба Онолын Үндэс

Odoo нь бүрэн модульчлагдсан систем бөгөөд модуль бүр тодорхой бизнесийн үйл ажиллагааг хариуцдаг. Жишээлбэл:

- hr — хүний нөөцийн бүртгэл
- purchase — худалдан авалт
- account — санхүүгийн бүртгэл
- portal — хязгаарлагдмал хэрэглэгчийн орчин

Модуль бүр дараах бүрэлдэхүүнүүдтэй байна:



Зураг 1.1: Odoo ERP модулийн бүтэц

- **models/** — өгөгдлийн бүтэц, бизнес логик
- **views/** — хэрэглэгчийн интерфейс
- **controllers/** — API болон веб хүсэлт
- **security/** — хэрэглэгчийн эрх ба хандалтын дүрэм

## 1. Судалгаа ба Онолын Үндэс

Энэхүү модульчлагдсан зохион байгуулалт нь системийг уян хатан, өргөтгөхөд хялбар болгож, байгууллагын өөрчлөлтөд дасан зохицох чадварыг нэмэгдүүлдэг.

### Техникийн орчин ба холболтын давхарга

Odoo ERP нь серверийн талд Python, өгөгдлийн талд PostgreSQL, веб серверийн хувьд Nginx эсвэл Odoo-ийн дотоод HTTP серверийг ашигладаг. Хэрэглэгчийн хүсэлт нь HTTP эсвэл JSON-RPC протоколоор Controller түвшинд ирж боловсруулагдана.

Үйлдвэрлэлийн орчинд (production environment) олон worker бүхий архитектур, Redis кэш, longpolling үйлчилгээ ашиглан системийн гүйцэтгэл, найдвартай байдлыг хангадаг.

## 1.3 Portal ба Controller

### Ерөнхий ойлголт

Odoo ERP системийн Portal нь байгууллагын дотоод мэдээллийн системийг хөнгөн хэрэглэгчийн орчинд нээх юм. Энэ нь байгууллагын гадаад болон дотоод хэрэглэгчдэд (жишээлбэл: харилцагч, нийлүүлэгч, ажилтан) Enterprise лиценз шаардахгүйгээр тодорхой өгөгдөл хандах, хүсэлт илгээх боломжийг олгодог.

Portal нь share хэрэглэгчийн эрхийн загварт тулгуурладаг бөгөөд res.users моделийн түвшинд share=True шинж чанараар ялгагдана. Энэ нь системийн "Internal User" төрлийн хэрэглэгчдээс ялгаатай, лицензийн тоонд хамаардаггүй боловч хэрэглэгчийн баталгаажсан (authenticated) төлөvt ажилладаг [1].

### Portal механизмийн бүтэц ба үүрэг

Odoo ERP системийн Portal орчин нь дараах гурван гол бүрэлдэхүүнээс бүрддэг:

- **Portal User (res.users.share=True)** — лицензгүй, портал хэлбэрийн хэрэглэгч. Тухайн хэрэглэгч "Portal" бүлэгт (base.group\_portal) харьялагдах бөгөөд зөвхөн өөрийн мэдээлэлд хандах хязгаарлалттай байдаг.
- **Portal Controller (Backend Layer)** — серверийн талын хяналтын давхарга бөгөөд хэрэглэгчийн хүсэлтүүдийг хүлээн авч, шалгаж, Model түвшний өгөгдлөтэй харьцаан, JSON хариу

## 1. Судалгаа ба Онолын Үндэс

буцаадаг.

- **Portal Interface (Frontend Layer – Next.js)** — орчин үеийн веб хэрэглэгчийн интерфэйс бөгөөд Odoo backend-тэй REST эсвэл JSON-RPC протоколоор холбогдон өгөгдлийг үзүүлэх, шинэчлэх, илгээх боломжийг олгодог.

Энэхүү бүтэц нь Odoo-ийн уламжлалт QWeb portal-ыг орчин үеийн вэб технологитой (React / Next.js) уялдуулсан илүү уян хатан архитектур бий болгож байгаа юм.

### Portal Controller-ийн архитектур

Odoo 18.0 хувилбарын Controller давхарга нь odoo.http.Controller дээр суурилдаг бөгөөд HTTP хүсэлт болон JSON RPC хүсэлтийг хулээн авч, бизнес логик (Model)-д хандалт хийнэ.

```
1 from odoo import http
2
3
4 class PortalLeaveController(http.Controller):
5
6     @http.route('/api/hr-leave/request', type='json', auth='user',
7                 methods=['POST'], csrf=False)
8
9     def create_time_request(self, **kwargs):
10
11         user = request.env.user
12
13         if not user.has_group('base.group_portal'):
14
15             return {'error': 'Access_denied'}
16
17         employee = request.env['hr.employee'].sudo().search([('user_id',
18
19             '=', user.id)], limit=1)
20
21         if not employee:
22
23             return {'error': 'Employee_not_found'}
24
25         leave = request.env['hr.leave'].sudo().create({
26
27             'employee_id': employee.id,
28
29             'name': kwargs.get('description'),
30
31             'request_date_from': kwargs.get('date_from'),
```

## 1. Судалгаа ба Онолын Үндэс

```
17     'request_date_to': kwargs.get('date_to'),  
18  
19     })  
  
    return {'success': True, 'leave_id': leave.id}
```

Код 1.1: Portal Controller-ийн жишээ

Энд auth='user' параметр ашиглагдаж байгаа нь портал хэрэглэгчийг баталгаажсан хэрэглэгчийн түвшинд шалгах бөгөөд дараа нь тухайн хэрэглэгч base.group\_portal бүлэгт багтсан эсэхийг шалгаж байна.

Ийм байдлаар auth='portal' гэх тусдаа төрөл шаардлагагүй болсон бөгөөд энэ нь Odoo 16+ хувилбаруудад authentication загвар шинэчлэгдсэнтэй холбоотой юм [3].

### Portal ба Next.js интерфэйсийн уялдаа

Next.js суурьтай портал интерфэйс нь Odoo-ийн backend API-д холбогдон хэрэглэгчийн өгөгдлөтэй харьцдаг. Хэрэглэгчийн талын (frontend) архитектур дараах дарааллаар ажиллана:

- **Login / Session Authentication** Хэрэглэгч нэвтэрсний дараа серверээс cookie эсвэл токен хүлээн авна.
- **API хүсэлт илгээх** Next.js талд fetch эсвэл axios ашиглан /api/... маршрут руу хүсэлт илгээнэ.
- **Backend Controller хүлээн авах** Controller нь auth='user' authentication шалгалт хийж, portal хэрэглэгч мөн эсэхийг group болон record rule-aap баталгаажуулна.
- **Model interaction** Controller нь request.env['model.name'] ашиглан бизнес логиктой харьцана.
- **Response буцаах** Controller нь JSON хэлбэрээр хариу өгч, Next.js UI хэсэгт шинэчилнэ.

```
1  await fetch("http://localhost:2050/api/hr-leave/request", {  
2      method: "POST",  
3      credentials: "include",
```

## 1. Судалгаа ба Онолын Үндэс

```
4     headers: { "Content-Type": "application/json" },  
5     body: JSON.stringify({ date_from: "2025-10-16", date_to: "  
6         2025-10-17", description: "Leave Request" })  
}) ;
```

Код 1.2: Next.js талын API хүсэлтийн жишиг

Энэхүү архитектур нь Odoo ERP системийг илүү API-центрик, фронтенд технологид нээлттэй болгож, хэрэглэгчийн туршлагыг сайжруулна.

### Аюулгүй байдлын давхарга

Portal хэрэглэгчид системийн өгөгдөл хандахдаа дараах олон давхар шалгалттай байдаг:

- **Authentication Layer:** auth='user' ашиглан session эсвэл токеноор баталгаажуулна.
- **Authorization Layer:** user.has\_group('base.group\_portal') шалгалт хийж зөвшөөрөгдсөн хэрэглэгчийг ялгана.
- **Access Control Layer:** ir.model.access.csv болон ir.rule файлуудаар тухайн хэрэглэгч зөвхөн өөрийн өгөгдөл хандахыг баталгаажуулна.

Ингэснээр portal хэрэглэгч системийн мэдээлэлд бүрэн хандахгүй боловч, өөрийн ажилтай холбоотой процессууд (цагийн хүсэлт, худалдан авалт, ирц бүртгэл гэх мэт) гүйцэтгэх боломжтой болдог. Эцэст нь portal болон Controller-ийн нэгдсэн механизм нь Odoo ERP системийн архитектурт нээлттэй, хөнгөн, өргөтгөх боломжтой давхарга бий болгодог.

## 1.4 Хандалтын эрхийн загвар(ir.model.access, ir.rule)

### Ерөнхий ойлголт

Odoo ERP систем нь өндөр түвшний эрхийн удирдлагын (Access Control) механизмаар дамжуулан хэрэглэгчдийн өгөгдөл хандах эрхийг нарийн хянадаг. Энэхүү эрхийн бүтэц нь хоёр үндсэн түвшинд хийгддэг:

## 1. Судалгаа ба Онолын Үндэс

- **Model access rule (`ir.model.access.csv`)** — хэрэглэгч тухайн модельд унших (read), бичих (write), үүсгэх (create), устгах (unlink) үйлдэл хийх эрхтэй эсэхийг хянадаг.
- **Record rule (`ir.rule`)** — тухайн модель доторх аль бичлэгүүдэд (records) хэрэглэгч хандаж болохыг тодорхойлдог.

Эдгээр нь Odoo ERP системийн security layer-ийн гол бүрэлдэхүүн бөгөөд portal хэрэглэгч системийн бизнес логик руу хязгаарлагдмал байдаа хандах нөхцөл бүрдүүлдэг.

### Онолын суурь: RBAC

Odoo-ийн эрхийн загвар нь үндсэндээ рольд суурилсан хандалтын удирдлага (RBAC) загварыг хэрэгжүүлдэг. RBAC нь хэрэглэгчдийг бүлэг/рольд хуваан, тухайн рольд оноосон эрхээр дамжуулан нөөцөд хандахыг зөвшөөрдөг [6]. `res.groups` (роль), `ir.model.access` (модель түвшний эрх), `ir.rule` (бичлэгийн түвшний домэйн) гурвын нэгдэл нь Odoo дахь RBAC-ийн практик хэрэгжилт болж, портал хэрэглэгчдэд ”зөвхөн өөрийнх”-ийг харах, ”зөвшөөрсөн талбар”-т л үйлдэх зэрэг хязгаарлалтыг баталгаажуулна.

### Portal хэрэглэгчийн хандалтын хязгаарлалт

Portal хэрэглэгч нь `share=True` шинжтэй хэрэглэгчийн төрөл бөгөөд системийн ”Internal User” төрлийн хэрэглэгчтэй харьцуулахад модель руу шууд хандах эрхгүй байдаг. Тодруулбал, portal хэрэглэгч Controller ашиглан REST API эсвэл JSON RPC замаар хүсэлт илгээсэн ч:

- Тухайн model дээр `group_portal` эсвэл түүнтэй холбогдох access rule байхгүй бол `AccessError` алдаа гарна.
- Portal хэрэглэгчийн `user_id` нь ”`base.group_user`” бүлэгт хамаарахгүй тул default security rule-үүд түүнд үйлчилдэггүй.

Иймд portal хэрэглэгчдэд зориулсан тусгай модель түвшний хандалтын файл (`ir.model.access.csv`) болон record rule (`security.xml`) зайлшгүй шаардлагатай байдаг.

### Model Access Rule (`ir.model.access.csv`)

## 1. Судалгаа ба Онолын Үндэс

ir.model.access.csv файл нь Odoo модулийн root түвшинд байрлах бөгөөд тухайн модульд ямар бүлгийн хэрэглэгч ямар модельд хандах эрхтэйг тодорхойлдог.

Жишээ (Portal хэрэглэгчийн хандалт үүсгэх):

```
1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,
  perm_unlink
2 access_hr_leave_portal,hr.leave,hr_holidays.model_hr_leave,base.
  group_portal,1,1,1,0
```

Код 1.3: ir.model.access.csv жишээ

Энэхүү тохиргоо нь:

- hr.leave моделийн хувьд портал хэрэглэгчид унших, бичих, үүсгэх (create) эрхийг олгоно,
- устгах (unlink) эрхийг хориглоно.

Ингэснээр Portal Controller нь request.env['hr.leave'].sudo() гэх мэтээр тухайн модельг дуудахад эрхийн алдаа үүсэхгүйгээр ажиллах боломжтой болно.

## Record Rule (ir.rule)

Model access rule нь хэрэглэгч тухайн модельд ямар төрлийн үйлдэл хийж болохыг хязгаарладаг бол record rule нь ямар бичлэгүүдэд (rows) хандаж болохыг тодорхойлдог.

Жишээ (Portal хэрэглэгч зөвхөн өөрийн бичлэгт хандах):

```
1 <record id="rule_hr_leave_portal_own" model="ir.rule">
2   <field name="name">Portal: Own Leave Only</field>
3   <field name="model_id" ref="hr_holidays.model_hr_leave"/>
4   <field name="groups" eval="[(4,ref('base.group_portal'))]"/>
5   <field name="domain_force">[('employee_id.user_id', '=', user.id)]<
    /field>
6 </record>
```

Код 1.4: ir.rule XML жишээ

## 1. Судалгаа ба Онолын Үндэс

Энэ нь дараах нөхцөлийг үүсгэнэ:

- Portal хэрэглэгч зөвхөн өөрийн employee\_id-тай холбоотой чөлөөний хүсэлтүүдийг (hr.leave) харах, шинэчлэх эрхтэй.
- Бусад ажилтны өгөгдөл рүү хандах боломжийг domain\_force нөхцлөөр бүрэн хориглоно.

### Controller ба Access Rule-ийн уялдаа

Portal Controller нь HTTP түвшинд хэрэглэгчийн хүсэлтийг хүлээн авч, тухайн хэрэглэгчийн environment (request.env)-ээр дамжуулан model-т ханддаг. Хэрэв portal хэрэглэгчийн хувьд тухайн model-д access line байхгүй бол:

AccessError: The requested operation cannot be completed due to security restriction  
гэсэн алдаа буцна.

Иймд controller зөв ажиллахын тулд portal хэрэглэгчийн эрхийн түвшинд:

- ir.model.access.csv – хэрэглэгчид model руу нэвтрэх үндсэн эрх олгох;
- ir.rule – тухайн хэрэглэгчийн бичлэгийг хязгаарлах нөхцөл (жишээлбэл user\_id = user.id) заах;

Гэсэн хоёр давхар тохиргоо шаардлагатай байдаг.

### Практик жишээ

Portal Controller кодын хэсэг:

```
1 @http.route('/api/hr-leave/request', type='json', auth='user', methods
2             =['POST'], csrf=False)
3
4     def create_time_request(self, **kwargs):
5         user = request.env.user
6
7         if not user.has_group('base.group_portal'):
8             return {'error': 'Access_denied'}
9
10        leave = request.env['hr.leave'].sudo().create({
```

## 1. Судалгаа ба Онолын Үндэс

```
8     'employee_id': request.env['hr.employee'].sudo().search([('
9         user_id', '=', user.id)], limit=1).id,
10    'name': kwargs.get('description'),
11    'request_date_from': kwargs.get('date_from'),
12    'request_date_to': kwargs.get('date_to'),
13 })
14
15     return {'success': True, 'leave_id': leave.id}
```

Код 1.5: Практик Controller кодын жишээ

Энэ controller зөв ажиллахын тулд дараах security файлууд шаардлагатай:

- `ir.model.access.csv` — base.group\_portal group-д `hr.leave` model-д CRUD хандалт олгох,
- `ir.rule` — зөвхөн тухайн хэрэглэгчийн (`user.id`) бичлэгүүдэд хандах нөхцөл үүсгэх.

## Дүгнэлт

Хандалтын эрхийн загвар нь Odoo ERP системийн security layer-ийн хамгийн чухал хэсэг юм. Portal хэрэглэгч Controller ашиглан API түвшинд model-т хандах үед, түүний эрх нь:

- `ir.model.access.csv` — үйлдлийн түвшин,
- `ir.rule` — бичлэгийн түвшин,

Гэсэн хоёр түвшинд хянаагддаг.

Ингэснээр:

- Portal хэрэглэгч Enterprise лицензгүйгээр тодорхой model-д хандалт хийж чадна,
- Мэдээллийн аюулгүй байдал хангагдана,
- Controller түвшинд эрхийн алдаа үүсэхгүйгээр CRUD үйлдэл гүйцэтгэх боломж бүрдэнэ.

Энэ шийдэл нь Portal механизмыг Enterprise лицензтэй системд зардал багатай, хөнгөн хувилбараар ашиглах түлхүүр шийдэл болдог.

## 2. ENTERPRISE СИСТЕМИЙН ЛИЦЕНЗ АШИГЛАЛТ

### 2.1 Байгууллагын ERP хэрэглээний өнөөгийн байдал (case-based судалгаа)

Odoo ERP-ийн Enterprise хувилбар нь нэг хэрэглэгчийн лицензээр бүх модульд хандах эрх олгодог бөгөөд хэрэглэгчийн тоо, ашиглах модулиудаас хамаарч лицензийн өртөг тооцогддог [5]. Төлбөртэй *internal user* бүрийг лицензийн сард/жилд тооцдог тул, аль модуль хэрэглэж байна вэ гэдгээс үл хамааран хэрэглэгчид бүх боломжийг авах боломжтой. Ийм зарчим нь бүрэн хэмжээний ERP хэрэгцээтэй ажилтнуудад тустай ч харьцангуй энгийн үйлдэл (цагийн хүсэлт илгээх, худалдан авалтын захиалга хийх, тайлан харах гэх мэт) хийх ажилтнуудад зардлыг нэмэгдүүлдэг. Одоогийн судалгаанд хамрагдсан Монголын жижиг, дунд бизнес эрхлэгчдийн төлөөлөл дээр тулгуурлахад дараах нийтлэг нөхцөл илэрсэн.

#### Хэрэглэгчийн лицензийн хэт хэрэглээ

- Ажилтан бүрд бүрэн эрх олгодог.** Олон байгууллага Enterprise лицензтэй учир ажилтнуудаа "internal user" болгон бүртгэж, хэрэглэгчийн тоогоор лицензийн төлбөр төлдөг. Гэтэл ажилчдын дийлэнх нь ERP-ийн цөөн функц (агаарын тээврийн замын хүсэлт, ирц бүртгэл, худалдан авалтын санал, тайлан хянах зэрэг) л ашигладаг бөгөөд энэ хэрэглээ нь Community хувилбарын чөлөөтэй функцээр хангалттай байж болох юм.
- Enterprise модуль бүрэн ашиглагддаггүй.** Enterprise хувилбар нь Studio, Helpdesk, Document Management, Finance, IoT гэх мэт олон нэмэлт модуль багтаадаг. Судалгаанд хамрагдсан компанийн ихэнх нь ердөө CRM, Sales, Purchase, Attendance зэрэг үндсэн модулиудыг ашиглаж, Enterprise хувилбарын нэмэлт модулиудыг ашигладаггүй. Энэ нь системийн үр ашигт байдал буурч, төлбөрийн үр ашиггүй байдлыг бий болгож байна.

## 2. ENTERPRISE СИСТЕМИЙН ЛИЦЕНЗ АШИГЛАЛТ

- **Лицензийг хугацаагаар сунгах үед зардлыг өсдөг.** Odoo Enterprise лиценз нь гэрээгээр тодорхой хугацаанд (жишээ нь 1, 2 эсвэл 3 жил) хийгдэх бөгөөд гэрээг сунгах үед нэг хэрэглэгчийн үнэ нэмэгдэх боломжтой [5]. Урт хугацааны гэрээ хийх, багц модулиудыг сонгох зэргээр өртөгийг бууруулах боломжтой боловч маш олон байгууллага үүнийг тооцоолдоггүй.

### Portal хэрэглэгч ашиглах боломж

Odoo-гийн лицензийн бодлогын дагуу portal болон public хэрэглэгчид төлбөргүй байдаг; зөвхөн системийн "internal" хэрэглэгчид лицензийн төлбөртэй ордог [5]. Гэхдээ олон байгууллага энэ боломжийг бүрэн ашигладаггүй. Portal нь дотоод ажилтан, нийлүүлэгч, харилцагч зэрэг гадны хэрэглэгчдэд зориулсан хялбар интерфэйс бөгөөд зөвхөн зөвшөөрсөн өгөгдөл дээр CRUD үйлдэл хийх боломжтой байдаг. Байгууллагын кейс судалгаагаар:

- **Цагийн хүсэлт.** Дотоод ажилтнууд ирц, цагийн хүсэлт гаргахдаа Enterprise хэрэглэгчийн лиценз авсан байх тохиолдол их. Энэ процесс Portal хэрэглэгчээр дамжуулан хэрэгжвэл лиценз шаардлагагүй бөгөөд зөвхөн хэрэгцээтэй мэдээлэлтэй ажиллах интерфэйсээр хязгаарлагдана.
- **Худалдан авалтын санал/захиалга.** Худалдан авалтын хүсэлт илгээх, нийлүүлэгчид санал тавих зэрэг энгийн процесийг portal-аар дамжуулж болно. Өөрөөр хэлбэл, зөвхөн "create" болон "read" эрхтэй portal user-ийг ашиглах замаар төлбөргүй хэрэглэгчээр худалдах процесс удирдах боломжтой.
- **Тайлан үзэх.** Удирдлагын түвшний эсвэл хариуцсан ажилтны хувьд тайлан харах шаардлага гарах үед системд шинээр license худалдан авахгүйгээр portal эрх үүсгэн тайланг хялбар интерфэйсээр үзүүлж болно.

Одоогийн нөхцөл байдлыг нэгтгэн дүгнэхэд, олон байгууллага Odoo Enterprise-ийг өргөнөөр ашиглаж байгаа боловч лицензийн хэт хэрэглээ болон Enterprise модуль ашиглалтын дутагдалтай

## 2. ENTERPRISE СИСТЕМИЙН ЛИЦЕНЗ АШИГЛАЛТ

байдал ажиглагддаг. Portal хэрэглэгчийн орчин, controller механизмыг ашигласнаар лицензтэй хэрэглэгчдийн тоог бууруулах, зардлыг хэмнэх, энгийн хэрэглэгчдэд зориулсан хөнгөн интерфэйсийг бий болгох боломжтой гэж үзэх үндэслэл бий.

### **Хэрэглээний кейсүүд (real-world examples)**

Монголын жижиг, дунд бизнесүүдийн хязгаарлагдмал мэдээлэл дээр үндэслэн дараах кейсүүдийг тэмдэглэж болно. **Кейс 1:**

Нэг компанид Odoo Enterprise 18 хувилбар хэрэгжүүлсэн. Компани нийт 50 ажилтантай бөгөөд тэднээс 15 нь санхүү, борлуулалт, худалдан авалт зэрэг модулиудыг бүрэн эрхээр хэрэглэх шаардлагатай байсан ч бүх 50 ажилтанд Enterprise лиценз олгожээ. Ингэснээр хэрэглэгчийн зардал 50 лицензийн төлбөрөөр тооцогдсон. Судалгааны явцад 35 ажилтан зөвхөн ирц бүртгэх, цагийн хүсэлт гаргах, хувь хүний мэдээлэл засварлах зэрэг үйлдэл л хийдэг нь илэрсэн; эдгээр хэрэглэгчдэд portal эрх хангалттай тул лицензийн тоог 50-аас 15 болгон бууруулах боломжтой байжээ.

### **Кейс 2:**

Нэгэн компанийн хувьд Odoo ERP 16 хувилбарыг нэвтрүүлсэн бөгөөд компанид 50 ажилтан ажилладаг ч Enterprise хувилбарын 20 дотоод (internal) хэрэглэгчийн лиценз худалдан авчээ. Эдгээрийн 5 лицензийг 35 хүн салбар салбартаа ээлжлэн ашиглаж, бараа материалын шаардах үүсгэх, худалдан авалтын захиалга гэх мэтийг бүртгэдэг байв. Энэ компани нэг нэвтрэх эрхийг олон ажилтан дамжуулан хэрэглэсэн нь аюулгүй байдалд эрсдэл үүсгэжээ. Дараа нь хэрвээ ямар нэг асуудал үүсэх бол хэн нь үүнийг хийсэн гэдэг нь мэдэгдэхгүй байх боломжтой байжээ.

Энэхүү хоёр кейсийн аль аль нь Enterprise лицензийн зардлыг бууруулах потенциалтай гэдгийг харуулж байна. Одоогоор олон байгууллага Odoo Enterprise лицензийг түлхүү хэрэглэж, портал хэрэглэгчийн орчныг бүрэн ашиглахгүй байгаа нь зардлын хэт өндөр байдалд хүргэж байна.

### **Дүгнэлт**

Байгууллагууд Odoo Enterprise системийг ашиглахдаа бүхий л ажилтнуудаа "internal

## 2. ENTERPRISE СИСТЕМИЙН ЛИЦЕНЗ АШИГЛАЛТ

"user" болгож лицензжүүлэх нь түгээмэл. Гэвч ажилтнуудын олонхи нь ERP-ийн цөөн функцтэй ажилладаг ба энэ нь portal user эрхээр хангагдах боломжтой. Odoo лицензийн бодлогын дагуу portal болон public users нь төлбөргүй байдаг, зөвхөн active internal users-ыг лицензийн төлбөрт оруулдаг [5]. Тиймээс portal механизмыг ашиглаж, хэрэглэгч бурд шаардлагатай эрхийг оновчтой тохируулах нь Enterprise лицензийн зардлыг бууруулж, системийн ашиглалтыг оновчтой болгох гол арга зам юм.

### 2.2 Лицензийн хүрээг багасгах

Энэ хэсэгт Odoo Enterprise-ийн лицензийн өргтийг бууруулах зорилгоор портал хэрэглэгчийн эрхийг ашиглах арга барил, түүний техникийн шийдэл, аюулгүй байдлын зохицуулалт, системийн архитектуртай уялдуулах загвар болон зардлын тооцооллыг танилцуулна. Агуулга нь өмнөх онолын хэсгүүд (Portal механизм, Controller, Access Rules)-т тулгуурласан практик хүрээтэй.

#### 2.2.1 *Portal хэрэглэгчийн бүтэц, эрхийн хязгаарлалт ба аюулгүй байдал*

Portal хэрэглэгч нь Odoo системд гадны буюу хөнгөн хэрэглэгчээр бүртгэгдэж, портал интерфэйсээр дамжуулан зөвхөн өөртэй нь хамаарах эсвэл түүнд хуваалцсан мэдээлэлд хандах эрхтэй аккаунт юм [1]. Odoo-ийн лицензийн бодлогоор идэвхтэй дотоод (internal) хэрэглэгч бүр төлбөртэйд тооцогддог бол, зөвхөн портал (share=True) хандалттай хэрэглэгчид лицензийн төлбөрөөс чөлөөлөгднө [5]. Иймээс байгууллагууд хязгаарлагдмал үүрэгтэй олон хэрэглэгчийг портал орчинд ажиллуулах замаар нийт лицензийн зардлаа бууруулах боломжтой [8, 9].

Портал хэрэглэгчид нь группээрээ base.group\_portal-д харьялгадж, анхдагч байдлаар ихэнх модел дээр унших төвтэй, хязгаарлагдмал эрхтэй байдаг. Ерөнхий зарчим нь "өөрийн мэдээлэл" хүрээнд харах эрх давамгайлах ба дурын өгөгдөл засвар хийх, устгах эрх нь хориглогддог. Энэ нь мэдээллийн аюулгүй байдлыг хамгаалах үндсэн давхарга болж өгдөг. Портал хэрэглэгчдийн ерөнхий зорилго нь дотоод ажилтнуудтай ижил бүрэн backend эрх эдлэх бус, тодорхой процессуудад оролцох хөнгөн орчныг хангахад оршино.

## 2. ENTERPRISE СИСТЕМИЙН ЛИЦЕНЗ АШИГЛАЛТ

### 2.2.2 *Controller, ACL, Record Rule ашиглан аюулгүй байдлыг хангах*

Портал орчинд өгөгдөл үүсгэх, хянах, хязгаарлахыг дараах гурван техник хэрэгслээр хэрэгжүүлнэ:

- **Веб Controller (Backend).** @`http.route` ашиглан портал хүсэлтийг хүлээн авч, зөвшөөрөгдсөн талбаруудыг валидац хийсний дараа загваруудтай харьцана. `auth='user'` ба `user.has_group('ba')` шалгалт нь портал хэрэглэгчдийг ялгана.
- **ACL — `ir.model.access.csv`.** Портал хэрэглэгч (`base.group_portal`)-д хэрэгтэй модел дээр туйлын хязгаарлагдмал `read/create` (шаардлагатай бол `write`) эрх олгоно; `unlink` ихэвчлэн хоригтой.
- **Record Rule — `ir.rule`.** ”Зөвхөн өөрийн бичлэг” гэх мэт домэйн нөхцлөөр бичлэгийн түвшний хандалтыг хязгаарлана (жишээ нь (`'employee_id.user_id', '=' , user.id`)). Ингэснээр нэг портал хэрэглэгч бусдын өгөгдөлд хандах боломжгүй.

### 2.2.3 *Бүлгийн дүгнэлт*

Портал хэрэглэгчийн эрхийг зөв зохион байгуулснаар Enterprise лицензийн хамаарлыг бууруулж, зардлын бүтцийг оновчлох боломжтой. Гол зарчим нь: (i) портал талд ”үүсгэх/илгээх/харах” зэрэг зайлшгүй шаардлагатай эрхийг ACL-ээр оновчтой олгох, (ii) Record Rule-оор зөвхөн өөрийн мэдээлэл рүүгээ хандах хязгаар тавих, (iii) контроллерийн түвшинд валидац, аудит, хамгаалалтын давхаргыг хангах. Ийм зохицуулалт нь зардал бууруулахын зэрэгцээ ERP ашиглалтын соёлыг төлөвшүүлж, мэдээллийн урсгалыг ил тод, хяналттай болгоно.

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

## 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

### 3.1 Системийн шаардлага

MW Portal системийн архитектур нь байгууллагын дотоод хэрэглэгчдэд зориулан Odoo ERP-ийн үндсэн функциуудыг портал орчинд хялбар, аюулгүй, найдвартай байдлаар хэрэгжүүлэх зорилготой. Системийн шаардлагыг функциональ ба функциональ бус гэж ангилж, дараах байдлаар индексжүүлэн тодорхойлсон.

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

#### 3.1.1 Функциональ шаардлагууд

Индекс	Шаардлагын нэр	Тайлбар
ФШ-100	Хэрэглэгчийн баталгаажуулалт	Хэрэглэгч Odoo-ийн эрхээр портал руу нэвтрэх, профайлын мэдээллийг харах, системээс гарах боломжтой байх.
ФШ-200	Ирцийн check-in, check-out, хийх	Хэрэглэгч ажлын ирцийн check-in, check-out бүртгэл хийх, сүүлийн ирцийн мэдээллийг харах функцтэй байх.
ФШ-300	Цагийн хүсэлт үүсгэх, харах	Хэрэглэгч цагийн төрлийг харах, шинэ хүсэлт үүсгэх, хүсэлтийн төлөвийг хянах боломжтой байх.
ФШ-400	Худалдан авалтын хүсэлт үүсгэх, харах	Хэрэглэгч худалдан авалтын шинэ хүсэлт үүсгэх, жагсаалт, дэлгэрэнгүй мэдээллийг харах, хүсэлтийг үргэлжлүүлэх буюу цуцлах боломжтой байх.
ФШ-500	Барааны шаардах хүсэлт үүсгэх, харах	Бүтээгдэхүүний зарлагын хүсэлт үүсгэх, холбогдох жагсаалт, гүйлгээний дэлгэрэнгүй мэдээллийг харах боломжтой байх.

Table 3.1: MW Portal системийн функциональ шаардлагууд

#### Системийн Функциональ Шаардлагуудын Нарийвчилсан Хүснэгт

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

<b>Функционал шаардлага ба дэд шаардлага</b>	<b>Тайлбар / Үйлдэл</b>	<b>Холбогдох API ба Frontend бүрэлдэхүүн</b>
<b>ФШ-100</b> <b>Хэрэглэгчийн баталгаажуулалт</b>	Хэрэглэгч Odoo-ийн эрхээр портал руу нэвтрэх, профайлын мэдээллийг харах, системээс гарах боломжтой байх.	
ФШ-101 — Системд нэвтрэх	Хэрэглэгч Odoo ERP эрхээр нэвтрэх.	POST /web/session/authenticate — login-form.tsx
ФШ-102 — Профайлын мэдээлэл харах	Нэвтэрсэн хэрэглэгчийн мэдээллийг харах.	GET /api/auth/employee-profile — Профайл бүрэлдэхүүн
ФШ-103 — Системээс гарах	Нэвтрэлт дуусгах.	POST /web/session/destroy — Logout товчлуур
ФШ-104 — Намайг санаарай	Cookie / localStorage ашиглах.	localStorage — login-form.tsx
<b>ФШ-200 Ирцийн удирдлага</b>	Ажилтан ирцийн check-in/out бүртгэл хийх, сүүлийн ирцийн мэдээлэл харах.	
ФШ-201 — Ирц бүртгэх	Check-in/out хийх.	POST /api/attendance/create — Ирц товчлуур

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

<b>Функционал шаардлага ба дэд шаардлага</b>	<b>Тайлбар / Үйлдэл</b>	<b>Холбогдох API ба Frontend бүрэлдэхүүн</b>
ФШ-202 — Ирцийн жагсаалт харах	Бүх ирцийн мэдээллийг харах.	GET /api/attendance/list — Ирцийн жагсаалт
ФШ-203 — Сүүлийн ирц харах	Хамгийн сүүлийн бүртгэл харах.	GET /api/attendance/list — Сүүлийн ирц хэсэг
<b>ФШ-300 Цагийн хүсэлт</b>	Хэрэглэгч цагийн төрлийг харах, хүсэлт үүсгэх, төлөв хянах.	
ФШ-301 — Цагийн төрлүүд харах	Боломжит төрлүүдийг харах.	GET /api/hr-leave/leave-types — Drop-down
ФШ-302 — Ургал харах	Ургал сонгох.	GET /api/hr-leave/leave-flow — Drop-down
ФШ-303 — Хүсэлт үүсгэх	Шинэ хүсэлт илгээх.	POST /api/hr-leave/time-request — Хүсэлт хэлбэр
ФШ-304 — Хүсэлтүүд харах	Өмнөх хүсэлтүүдийн жагсаалт.	GET /api/hr-leave/time-requests — Жагсаалт бүрэлдэхүүн

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

<b>Функционал шаардлага ба дэд шаардлага</b>	<b>Тайлбар / Үйлдэл</b>	<b>Холбогдох API ба Frontend бүрэлдэхүүн</b>
ФШ-305 — Хүсэлтийн төлөв хянах	Хүсэлтийн явцыг харах.	GET <code>/api/hr-leave/time-requests</code> — Төлөвийн хэсэг
<b>ФШ-400 Худалдан авалт</b>	Худалдан авалтын хүсэлт үүсгэх, жагсаалт, дэлгэрэнгүй харах.	
ФШ-401 — Хүсэлт үүсгэх	Шинэ худалдан авалтын хүсэлт илгээх.	POST <code>/api/purchase-request/create</code> — Үүсгэх хэлбэр
ФШ-402 — Хүсэлтүүд харах	Бүх хүсэлтийн жагсаалт.	GET <code>/api/purchase-request</code> — Жагсаалт
ФШ-403 — Дэлгэрэнгүй харах	Тухайн хүсэлтийн дэлгэрэнгүй.	GET <code>/api/purchase-request/&lt;id&gt;</code> — Дэлгэрэнгүй хуудас
ФШ-404 — Үргэлжлүүлэх шат	Шат дамжуулах.	POST <code>/api/purchase-request/&lt;id&gt;/action-next</code> — “Үргэлжлүүлэх” товч
ФШ-405 — Цуцлах	Хүсэлтийг цуцлах.	POST <code>/api/purchase-request/&lt;id&gt;/action-cancel</code> — “Цуцлах” товч

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

<b>Функционал шаардлага ба дэд шаардлага</b>	<b>Тайлбар / Үйлдэл</b>	<b>Холбогдох API ба Frontend бүрэлдэхүүн</b>
ФШ-406–410 — Нэмэлт өгөгдлүүд авах	Салбар, агуулах, урсгал, приорити, бүтээгдэхүүн авах.	GET <a href="/api/purchase-request/">/api/purchase-request/...</a> — Drop-down, сонгох хэсэг
<b>ФШ-500 Барааны шаардах</b>	Бүтээгдэхүүний зарлагын хүсэлт, жагсаалт, дэлгэрэнгүй харах.	
ФШ-501 — Шинэ хүсэлт үүсгэх	Зарлагын хүсэлт үүсгэх.	POST <a href="/api/product-expense/create">/api/product-expense/create</a> — Үүсгэх хэлбэр
ФШ-502 — Хүсэлтүүд харах	Бүх хүсэлтүүдийн жагсаалт.	GET <a href="/api/product-expense">/api/product-expense</a> — Жагсаалт
ФШ-503 — Дэлгэрэнгүй харах	Нэг хүсэлтийн дэлгэрэнгүй.	GET <a href="/api/product-expense/&lt;id&gt;">/api/product-expense/&lt;id&gt;</a> — Дэлгэрэнгүй хуудас
ФШ-504–510 — Урсгал, цуцлах, сонголтууд	Агуулах, урсгал, гүйлгээний утга, бүтээгдэхүүн авах.	GET <a href="/api/product-expense/">/api/product-expense/...</a> — Drop-down хэсэг

Table 3.2: MW Portal системийн функциональ шаардлагуудын нэгтгэсэн хүснэгт

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

#### 3.2 Ажлын явц

Доорх хүснэгтүүд нь MW Portal системийн функциональ шаардлагын дагуу бүртгэгдсэн үндсэн хэрэглээний тохиолдлууд (Use Case)-ыг илэрхийлнэ. Эдгээр нь хэрэглэгчийн гүйцэтгэх үйлдлийн зорилго, урьдчилсан нөхцөл, амжилттай ба бүтэлгүйтлийн төгсгөл, үйл явцын алхмуудыг тодорхой заасан болно.

##### **ФШ-10: Хэрэглэгчийн баталгаажуулалт**

Талбар	Тайлбар
Use Case нэр	Хэрэглэгчийн баталгаажуулалт
Зорилго (Goal)	Хэрэглэгч системд Odoo эрхээр нэвтрэх, хувийн мэдээлэл харах, системээс гарах боломжтой байх
Үтвар нөхцөл (Pre-condition)	Хэрэглэгчдэд Odoo-н эрх үүссэн, порталд нэвтрэх хаяг байгаа
Амжилттай төгсгөл (Success Condition)	Хэрэглэгч амжилттай нэвтрэх, профайлаа харах, системээс гарах үйлдлийг хийж чадах
Бүтэлгүйтлийн нөхцөл (Failure Condition)	Нууц үг буруу оруулах, эрх блоклогдсон байх, систем хариу өгөхгүй байх
Үндсэн тоглогч (Primary Actor)	Ажилтан (Портал хэрэглэгч)
<i>Үргэлжлэл...</i>	

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

**Table 3.3 – Үргэлжлэл**

Талбар	Тайлбар
<b>Үндсэн үйл явц (Main Flow)</b>	<ol style="list-style-type: none"> <li>Хэрэглэгч логин хэсэг рүү орно.</li> <li>Хэрэглэгч и-мэйл, нууц үгээ оруулна.</li> <li>Систем эрхийг шалгаж, нэвтрүүлнэ.</li> <li>Хэрэглэгч профайл мэдээллээ хянах боломжтой болно.</li> <li>”Logout” хийж системээс гарна.</li> </ol>
<b>Өргөтгөл (Extension)</b>	Байхгүй
<b>Хувилбарууд (Alternative Flow)</b>	Нууц үг мартсан тохиолдолд ”Forgot Password” урсалаар сэргээх

#### **ФШ-20: Ирцийн удирдлага**

Талбар	Тайлбар
<b>Use Case нэр</b>	Ирц бүртгэх
<b>Зорилго (Goal)</b>	Хэрэглэгч ажилдаа ирсэн болон тарсан хугацааг бүртгэх
<b>Үгтвар нөхцөл (Pre-condition)</b>	Хэрэглэгч системд амжилттай нэвтэрсэн байна
<b>Амжилттай төгсгөл (Success Condition)</b>	Ирцийн бүртгэл үүсэж, системд хадгалагдсан байх
<i>Үргэлжлэл...</i>	

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

Table 3.4 – Ургэлжлэл

Талбар	Тайлбар
<b>Бүтэлгүйтлийн нөхцөл (Failure Condition)</b>	Давхар бүртгэл, интернет холболтгүй, системийн алдаа
<b>Үндсэн тоглогч (Primary Actor)</b>	Ажилтан (Портал хэрэглэгч)
<b>Үндсэн үйл явц (Main Flow)</b>	<ol style="list-style-type: none"> <li>1. Хэрэглэгч ”Ирц” хэсэг рүү орно.</li> <li>2. ”Check-in” товчийг дарна.</li> <li>3. Ажлын цаг дуусахад ”Check-out” товч дарна.</li> <li>4. Ирцийн төлөв харагдана.</li> </ol>
<b>Өргөтгөл (Extension)</b>	Байхгүй
<b>Хувилбарууд (Alternative Flow)</b>	Автомат check-out нөхцөл (урьдчилан тохируулсан үед)

### ФШ-30: Цагийн хүсэлт удирдах

Талбар	Тайлбар
<b>Use Case нэр</b>	Цагийн хүсэлт илгээх
<b>Зорилго (Goal)</b>	Хэрэглэгч чөлөө авах, илүү цагаар ажиллах хүсэлт гаргах
<b>Үтвар нөхцөл (Pre-condition)</b>	Системд нэвтэрсэн байна
<i>Ургэлжлэл...</i>	

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

Table 3.5 – Үргэлжлэл

Талбар	Тайлбар
<b>Амжилттай төгсгөл (Success Condition)</b>	Хүсэлт системд амжилттай үүсэж, төлөв харагдах
<b>Бүтэлгүйтлийн нөхцөл (Failure Condition)</b>	Дутуу мэдээлэл оруулах, систем хариу өгөхгүй байх
<b>Үндсэн тоглогч (Primary Actor)</b>	Ажилтан (Портал хэрэглэгч)
<b>Үндсэн үйл явц (Main Flow)</b>	<ol style="list-style-type: none"> <li>”Цагийн хүсэлт” хэсэг рүү орно.</li> <li>Хүсэлтийн төрөл болон хугацааг сонгоно.</li> <li>”Илгээх” товч дарна.</li> <li>Төлөвийг хянана.</li> </ol>
<b>Өргөтгөл (Extension)</b>	Байхгүй
<b>Хувилбарууд (Alternative Flow)</b>	Байхгүй

**ФШ-40: Худалдан авалтын хүсэлт үүсгэх**

Талбар	Тайлбар
<b>Use Case нэр</b>	Худалдан авалтын хүсэлт
<i>Үргэлжлэл...</i>	

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

**Table 3.6 – Үргэлжлэл**

Талбар	Тайлбар
<b>Зорилго (Goal)</b>	Ажилтан шаардлагатай бараа/үйлчилгээний худалдан авалтад хүсэлт гаргах
<b>Үгтвар нөхцөл (Pre-condition)</b>	Системд нэвтэрсэн байна
<b>Амжилттай төгсгөл (Success Condition)</b>	Хүсэлт амжилттай үүсэж, жагсаалт болон дэлгэрэнгүй мэдээлэл харагдах
<b>Бүтэлгүйтлийн нөхцөл (Failure Condition)</b>	Бүтээгдэхүүн сонгоогүй, салбар дутуу, системийн алдаа
<b>Үндсэн тоглогч (Primary Actor)</b>	Ажилтан (Портал хэрэглэгч)
<b>Үндсэн үйл явц (Main Flow)</b>	<ol style="list-style-type: none"> <li>”Худалдан авалт” хэсэг рүү орно.</li> <li>Бүтээгдэхүүн, тоо хэмжээ, салбарыг сонгоно.</li> <li>”Хүсэлт илгээх” товч дарна.</li> <li>Хүсэлтийн төлөвийг хянаж үзнэ.</li> </ol>
<b>Өргөтгөл (Extension)</b>	Байхгүй
<b>Хувилбарууд (Alternative Flow)</b>	Байхгүй

**ФШ-50: Барааны зарлагын хүсэлт**

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

Талбар	Тайлбар
<b>Use Case нэр</b>	Бараа материалын зарлага гаргах хүсэлт
<b>Зорилго (Goal)</b>	Эмийн сангаас эсвэл агуулахаас бараа зарлагдах хүсэлт гаргах
<b>Үгтвар нөхцөл (Pre-condition)</b>	Системд амжилттай нэвтэрсэн байна
<b>Амжилттай төгсгөл (Success Condition)</b>	Хүсэлт амжилттай үүсэж, агуулахад хүргэгдэнэ
<b>Бүтэлгүйтлийн нөхцөл (Failure Condition)</b>	Бүтээгдэхүүн дутуу, санд бүртгэлгүй, системийн алдаа
<b>Үндсэн тоглогч (Primary Actor)</b>	Ажилтан (Портал хэрэглэгч)
<b>Үндсэн үйл явц (Main Flow)</b>	<ol style="list-style-type: none"> <li>”Барааны зарлага” хэсэг рүү орно.</li> <li>Зарлагдах бараа, тоо хэмжээ оруулна.</li> <li>”Хүсэлт илгээх” товч дарна.</li> <li>Хүсэлтийн төлөвийг хянаж үзнэ.</li> </ol>
<b>Өргөтгөл (Extension)</b>	Байхгүй
<b>Хувилбарууд (Alternative Flow)</b>	Байхгүй

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

#### 3.3 Системийн архитектурын ерөнхий бүтэц

MW Portal систем нь Odoo ERP-ийн Enterprise орчинд портал хэрэглэгчийн оролцоог нэмэгдүүлэн, системийн нийт зардлыг оновчтой байлгах зорилготойгоор архитектурын зохион байгуулалт хийгдсэн. Архитектурын гол онцлог нь техникийн бүтцийг бизнесийн бодит хэрэглээтэй уялдуулсан, хөнгөн, өргөтгөгдөхүйц байхаас гадна аюулгүй байдал, найдвартай ажиллагааг хангахуйц байхаар төлөвлөгдсөнд оршино.

##### 3.3.1 Архитектурын үндсэн санаа

Судалгааны явцад илэрсэн гол асуудал нь байгууллагууд ERP системийг нэвтрүүлэхдээ зөвхөн лицензтэй хэрэглэгчид дамжуулан системд хандах боломжтой тул шаардлагагүй лицензийн зардал үүсч, нэг хэрэглэгчийн эрхийг олон хүн хуваан ашиглах нөхцөл байдал үүсэж байв. Энэ нь audit trail, аюулгүй байдлын хувьд эрсдэлтэйгээс гадна системийн хэрэглээг буруу зуршил руу хөтөлдөг байна.

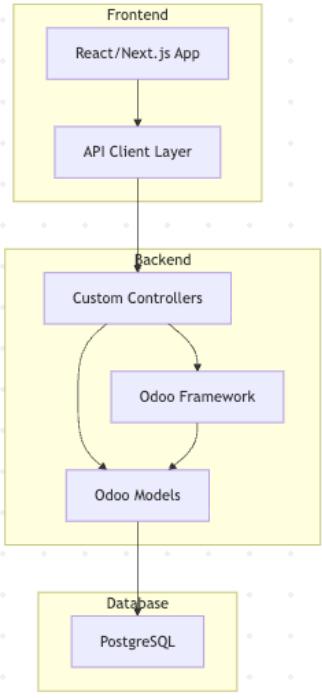
Иймд архитектурын үндсэн шийдэл нь ”Portal хэрэглэгч” нэртэй хязгаарлагдмал эрхтэй хэрэглэгчдийг тусгай интерфэйсээр холбож, зөвхөн тодорхой функцийн хандаж боломж олгож, үндсэн ERP системийн ачаалал болон лицензийн өртгийг бууруулахад чиглэгдсэн юм.

##### 3.3.2 Архитектурын бүтэц

Системийн бүтэц нь гурван үндсэн давхаргад хуваагдана. Үүнд:

- Portal хэрэглэгчийн веб интерфэйс — Next.js дээр суурилсан орчин үеийн UI
- MW API backend — Python, Odoo Framework ашигласан дунд түвшний логик
- ERP системийн өгөгдлийн давхарга — Odoo-ийн model болон PostgreSQL өгөгдлийн сан

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР



Зураг 3.1: MW Portal системийн ерөнхий архитектур

Portal хэрэглэгч нь системд нэвтэрсний дараа өөрийн хариуцсан мэдээллийг илгээх, шалгах, хүсэлт гаргах зэрэг үйлдлийг UI-аар дамжуулан MW Backend API руу илгээдэг. API нь хэрэглэгчийн эрхийг шалгаж, Odoo-ийн дотоод бизнес логиктой харьцан мэдээллийг зохистойгоор дамжуулдаг.

#### 3.3.3 Архитектурын дизайны шийдвэрүүд

##### Portal хэрэглэгчийн эрхийн загвар

Portal хэрэглэгчид нь Odoo ERP-ийн `base.group_portal` бүлэгт хамаарах бөгөөд зөвхөн өөрт хамаарах өгөгдлийг харах, шинээр үүсгэх зэрэг эрхтэй. Хандалтын хяналтыг `ir.model.access` болон `record_rule`-ээр нарийн тогтоосон ба энэ нь дотоод өгөгдөл рүү зөвшөөрөлгүй хандалтаас сэргийлэх найдвартай аргачлал болдог.

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

#### UI ба API-н салангид бүтэц

Next.js ба React суурьтай UI нь ERP системийн дотоод интерфэйсээс бүрэн тусгаарлагдсан. Энэ нь хэрэглэгчийн туршлагыг хөнгөн, хурдан, тасралтгүй байлгахын зэрэгцээ, ачааллыг систем хооронд зохицуулж, аюулгүй байдлыг хангах нөхцөл бүрдүүлдэг. Backend талд custom логик бүхий Python код ашиглан ERP рүү дамжих хүсэлтүүдийг удирдана.

##### 3.3.4 Судалгаанд тулгуурласан ажлын урсгал

Судалгаанд хамрагдсан байгууллагуудын хувьд портал хэрэглэгчид дараах төрлийн хүсэлт, мэдээллийг хамгийн өндөр давтамжтайгаар ERP-д илгээдэг болох нь ажиглагдсан. Тиймээс архитектурын загвар эдгээр урсгалыг дэмжих байдлаар тусгайлан төлөвлөгдсөн:

№	Урсгалын нэр	API Endpoint	Odoο Модель
1	Ирц бүртгэх	/api/attendance/check_in	hr.attendance
2	Цагийн хүсэлт	/api/hr-leave/time-request	hr.leave.mw
3	Худалдан авалт	/api/purchase-request/create	purchase.request
4	Барааны шаардах	/api/product-expense/create	stock.product.other.expense

Table 3.8: Portal хэрэглэгчийн түгээмэл урсгалуудын API холболт ба Odoο модель

Дээрх урсгал тус бүрт портал хэрэглэгч зөвхөн "create" (үүсгэх) үйлдлийг хийх ба эдгээр хүсэлт нь дотоод хэрэглэгч буюу менежерийн түвшинд батлагдан үргэлжилдэг.

#### 3.4 Системийн архитектурын нарийвчилсан бүтэц

MW Portal систем нь олон давхартгат архитектурт суурилсан бөгөөд хэрэглэгчийн UI түвшнээс өгөгдлийн санд хүргэх бүхий л үйлдлийг тодорхой дараалал, дүрэм, хяналтын механизмаар боловсруулдаг. Энэхүү нарийвчилсан архитектур нь зөвхөн техникийн төдийгүй хэрэглээний

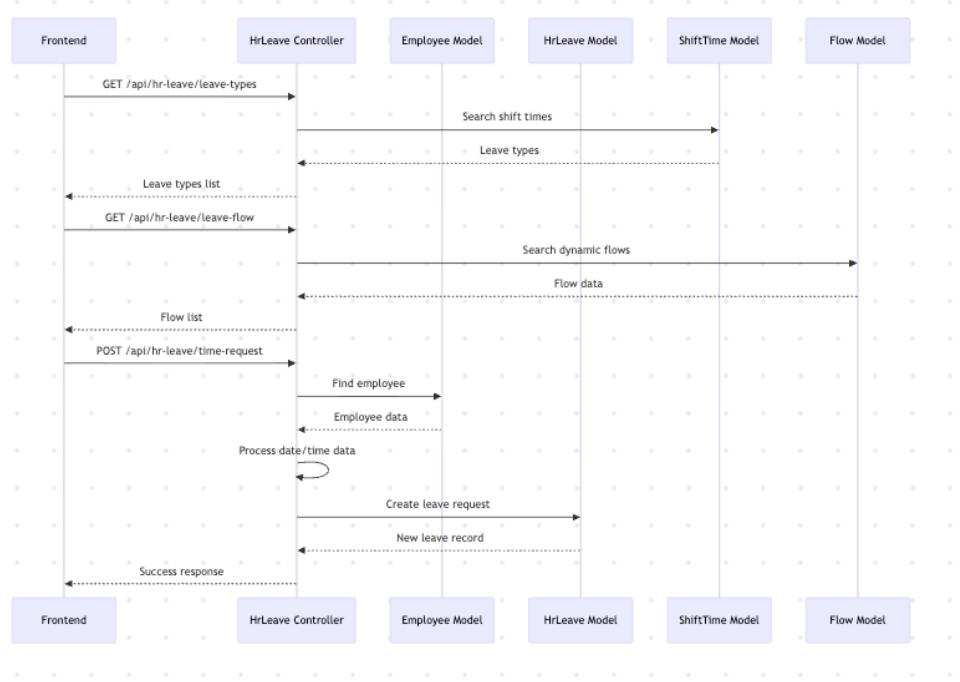
### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

онцлог, бизнесийн хэрэгцээ, мэдээллийн аюулгүй байдал зэргийг цогцоор нь хамарсан шийдэл юм.

#### 4.2 Sequence диаграмм ба бизнес урсгал

Системд хэрэглэгчээс ирэх үйлдэл бүр тодорхой дараалалтайгаар боловсруулагддаг бөгөөд үүнийг дараах диаграммуудаар тайлбарлана:

**Чөлөөний хүсэлт үүсгэх урсгал:** Хэрэглэгч өөрийн shift болон төрлөөс хамаарсан чөлөөний хүсэлт илгээхэд, систем динамик flow-г шалгаж, тухайн ажилтантай холбоотой мэдээллийг Odoo моделуудаас цуглуулж, шинэ хүсэлт үүсгэдэг.



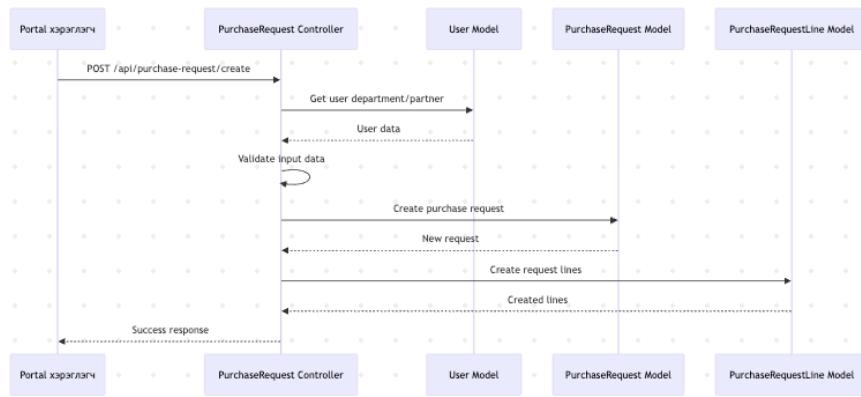
Зураг 3.2: Чөлөөний хүсэлт үүсгэх урсгалын дарааллын диаграмм

**Тайлбар:** Хэрэглэгчийн хүсэлтийн үеэр систем нь ажилтны мэдээллийг Employee модель дээр үндэслэн шүүж, зөвхөн тухайн ажилтанд хамаарах боломжит чөлөөний төрлийг харуулдаг.

**Худалдан авалтын хүсэлт илгээх урсгал:** Portal хэрэглэгч байгууллагын салбар, агуулах зэргийг суурь параметр болгон ашиглан худалдан авалтын хүсэлт үүсгэж, тухайн хүсэлтийн

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

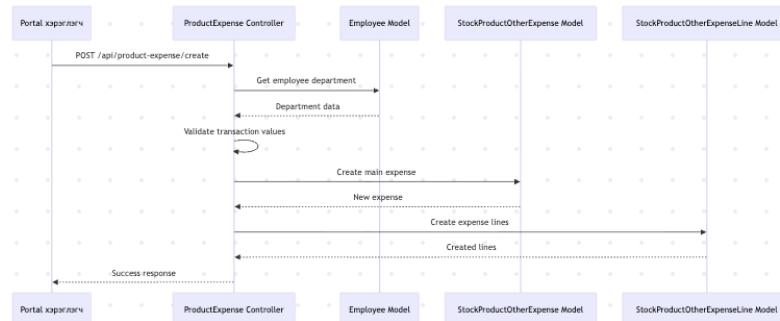
мөр бүрийг Odoo модельд холбон бүртгэдэг.



Зураг 3.3: Худалдан авалтын хүсэлт үүсгэх урсгалын дарааллын диаграмм

**Тайлбар:** Хүсэлтийн мөр бүр өөрийн нэр төрөл, тоо хэмжээ, үнэ гэх мэт өгөгдлийг шүүлтүүрээр шалгаж, Purchase Request болон түүний мөрийн модельд (Lines) бүртгэгддэг.

**Бараа материалын зардал үүсгэх урсгал:** Нэг төрлийн зардлыг олон төрлийн бүтээгдэхүүнээр илэрхийлж болох бөгөөд хэрэглэгчийн харьяа салбар, алба дээр үндэслэн тайлан үүсгэгдэнэ.



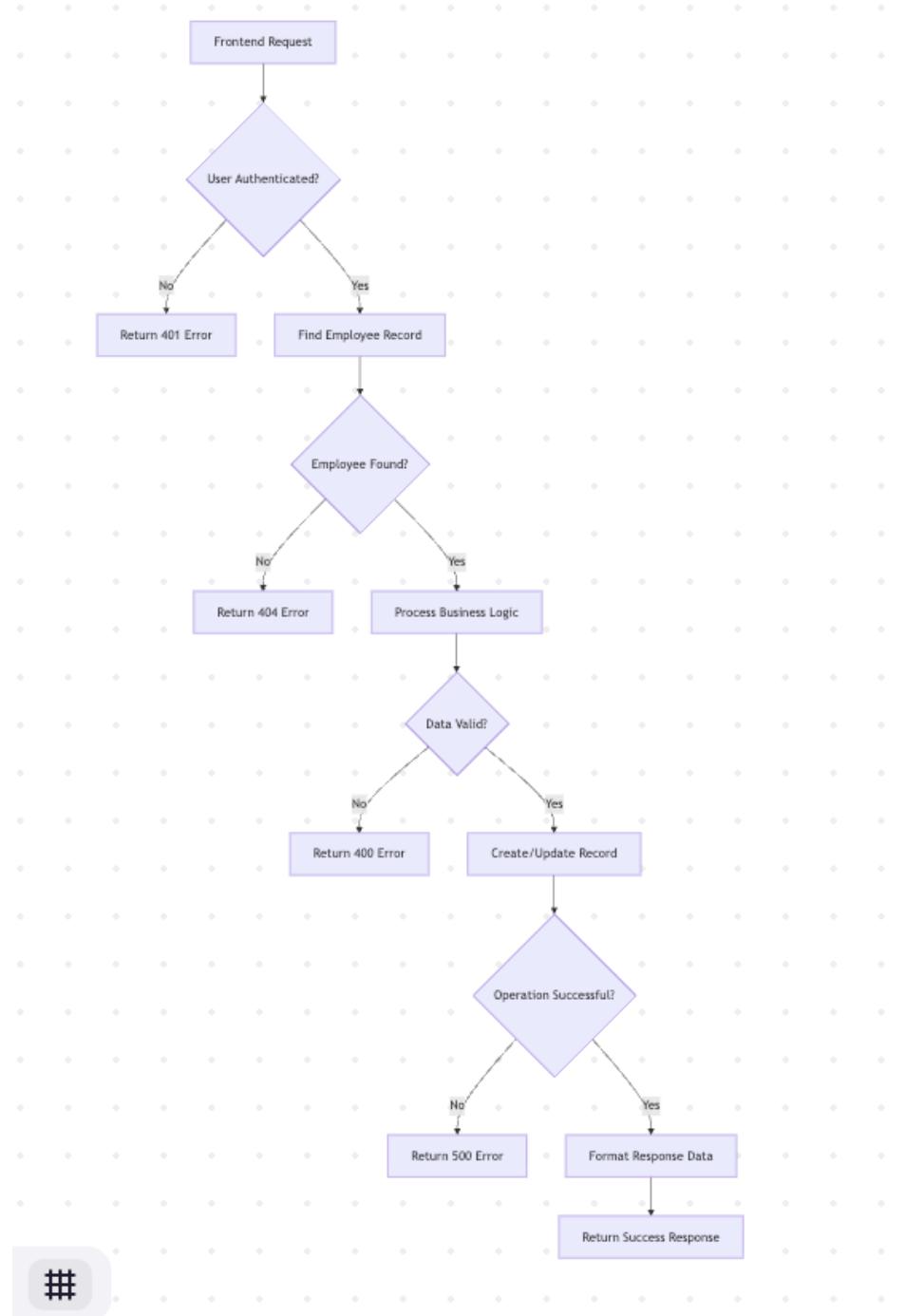
Зураг 3.4: Бараа материалын зарлага үүсгэх урсгалын дарааллын диаграмм

**Тайлбар:** Систем нь тухайн ажилтны хамааралтай салбарын мэдээллийг Employee модельд эээр үндэслэн тодорхойлж, зарлагын бүртгэлийг үндсэн болон мөрийн түвшинд (expense + expense line) үүсгэн хадгалдаг.

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

#### 4.3 Өгөгдлийн урсгал ба шалгалтын процесс

Бүх API дуудлага дараах ерөнхий шалгалтын бүтэцтэй байна:



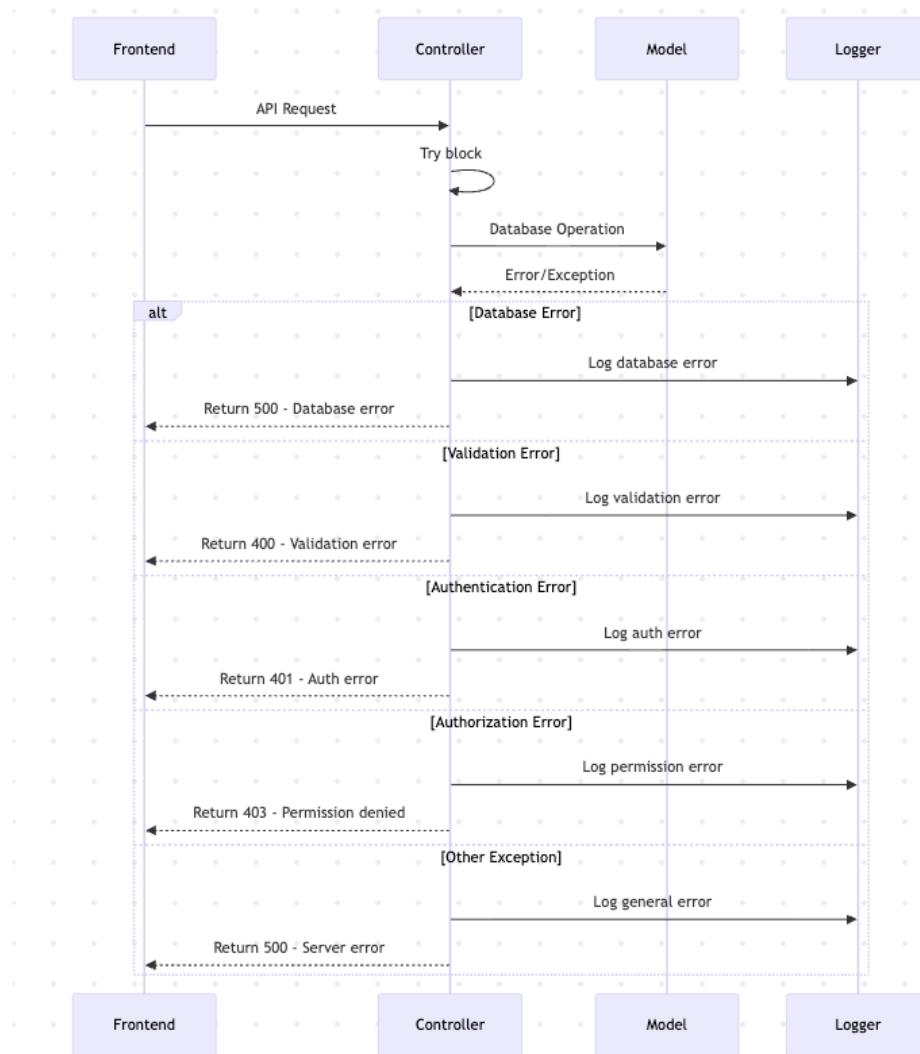
Зураг 3.5: Өгөгдлийн урсгал ба шалгалтын процесийн диаграмм

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

**Тайлбар:** Хэрэглэгчийн authentication (нэвтрэлтийн эрх), employee record шалгалт, өгөгдлийн форматын зөв эсэх зэргийг шат дараатай шалгаж, шаардлага хангаагүй тохиолдолд тодорхой алдааны хариу буцаадаг. Харин өгөгдөл зөв бол моделиудад мэдээлэл бүртгэгдэж, амжилттай хариу өгнө.

#### 4.4 Алдааны боловсруулалт

Систем нь алдааг ангилан бүртгэж, тодорхой HTTP кодоор хариу өгдөг. Энэ нь хөгжүүлэлт болон хэрэглэгчийн зүгээс асуудлыг илрүүлэхэд хялбар болгодог:



Зураг 3.6: Алдааны боловсруулалтын диаграмм

### 3. СИСТЕМИЙН ЗОХИОМЖ АРХИТЕКТУР

**Тайлбар:** Шалгалт бүрийн үе шатанд гарч болох authentication, validation, permission, database зэрэг төрлийн алдааг ялган боловсруулж, серверийн log бүртгэлд бүртгэнэ. Ингэснээр системийн хэвийн ажиллагаа болон аюулгүй байдлыг хангах нөхцөл бүрддэг.

# Bibliography

- [1] Odoo S.A. (2024). Odoo Enterprise Licensing Guide 18.0. Odoo Official Documentation. Retrieved from <https://www.odoo.com/documentation/18.0/legal/licenses.html>
- [2] Monk, C., & Wagner, T. (2023). Enterprise Resource Planning Systems: Concepts and Practice. Springer.
- [3] Odoo S.A. (2024). Odoo 18.0 Documentation — Developer: ORM and Architecture Overview. Odoo Official Documentation. Retrieved from [https://www.odoo.com/documentation/18.0/developer/tutorials/server\\_framework\\_101/01\\_architecture.html](https://www.odoo.com/documentation/18.0/developer/tutorials/server_framework_101/01_architecture.html)
- [4] Odoo S.A. (2024). Odoo 18.0 Developer Documentation — Security, Access Rules, and Record Rules. Odoo Official Documentation. Retrieved from <https://www.odoo.com/documentation/18.0/developer/reference/backend/security.html>
- [5] Ventor. (n.d.). Odoo Pricing. Retrieved from <https://ventor.tech/odoo-pricing/>
- [6] Sandhu, R., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. IEEE Computer, 29(2), 38–47.
- [7] OWASP Foundation. (2023). OWASP Application Security Verification Standard 4.0.3. Retrieved from <https://owasp.org/www-project-application-security-verification-standard/>
- [8] Polskaya, I., & Puntus, V. (2023). Odoo Enterprise лицензийн үнэ ба зардлыг бууруулах арга зам. VentorTech Blog. Retrieved from <https://ventor.tech/odoo/odoo-enterprise-license-pricing-and-how-to-reduce-cost/>
- [9] Odoo S.A. (2025). Odoo Enterprise захиалгын гэрээ (Үйлчилгээний нөхцөл, хувилбар 11). Odoo 19.0 Documentation. Retrieved from <https://www.odoo.com/documentation/19.0/legal/terms/enterprise.html>
- [10] Cudio. (2025). Odoo ERP нэвтрүүлэлтийн зардал: Үнэ, далд зардал ба төсвийн зөвлөмж. Cudio Tech Blog. Retrieved from <https://www.cudio.com/blogs/odoo-erp-implementation>

## **А. СУДАЛГААНЫ ХҮСНЭГТ, ЛИЦЕНЗИЙН ХАРЬЦУУЛАЛТ**

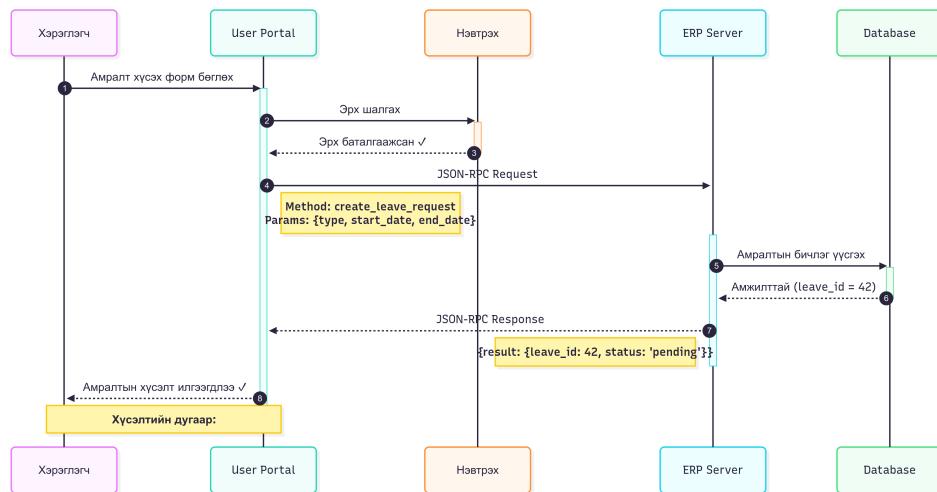
Хавсралтын агуулга

## B. ODOO ACCESS RULE ЖИШЭЭ

Хавсралтын агуулга

## C. PORTAL CONTROLLER-ИЙН КОДЫН ХЭСЭГ (ЖИШЭЭ ХЭЛБЭРЭЭР)

Хавсралтын агуулга



Зураг C.1: Портал хэрэглэгчийн use case — хүсэлт илгээх ба дотоод урсгалтай уялдах жишээ

- `models/` — өгөгдлийн бүтэц, бизнес логик
- `views/` — хэрэглэгчийн интерфэйс
- `controllers/` — API болон веб хүсэлт
- `security/` — хэрэглэгчийн эрх ба хандалтын дүрэм