

ЗМІСТ

					ДП.ПЗ 04.00.00.000 ПЗ							
Зм.	Арк.	№ докум.	Підпис	Дата	Розробка програмного алгоритмічного забезпечення багатофункціональної корпоративної системи для сумісної роботи, управління документами і проектами ПОЯСНЮВАЛЬНА ЗАПИСКА	Літ.		Аркуш	Аркушів			
Розробив	Бойчук Я.В.							1	??			
Перевірів	Випасняк Л.І.					ІНФТУНГ ПЗ-07-1						
Т.контр												
Н.контр												
Затв.												

ВСТУП

В нас час важливою економічною точкою опори для будь-якої комерційної організації є наявність певних факторів, які визнають чітку позицію компанії на ринку. До всіх цих чинників можна віднести багато варіантів, зокрема:

- а) капітал підприємства;
- б) матеріальна база;
- в) технічна база;
- г) кваліфікований персонал;
- д) місце підприємства на внутрішньому і зовнішньому ринку;
- е) наявність сучасних засобів виробництва та ведення бізнесу;
- ж) та інші.

Тому кожний розділ ведення бізнесу повинний бути детально розглянутий та впроваджений у життя. Але якщо подивитися із точки зору програмного забезпечення, то на даному етапі розвитку цивілізації, якісне ПЗ відіграє напевно найбільш важливу роль. Адже не можливо зараз утримувати всі дані в паперовому вигляді, не можливо відсилати друковані листи, чи спілкуватися тільки по телефоні і взнавати новини компанії тільки при зустрічі. У наш стрімкий час розвитку, новини міняються із колосальною швидкістю, тому встигнути за всім просто не можливо без певного програмного продукту. Уявіть собі інформатор, який сповіщає будь-які для Вас новини чи корисну інформацію в зручний для Вас час, при цьому вміє фільтрувати і аналізувати дані із попередніх запитів. Також на даний момент важко уявити не можливість спільної роботи над документами, над електронними таблицями. Дані технології вже давно використовуються людьми і підприємствами, починаючи від найменших де працює двоє людей, до величезних із кількістю працівників більше ста тисяч. Але для цього всього використовуються дуже багато технологій, які важко налаштувати і потребують великих витрат на

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		2

підтримку. Тому було розроблено багато сервісів і додатків, які полегшують роботу в мережі для підприємств.

Дане ПЗ використовується у всіх нішах нашого життя, починаючи від шкіл і лікарень, закінчуючи величезними корпораціями з будівництва космічних кораблів. Тому розробити універсальний продукт, який забезпечить всі вимоги, просто не можливо. Для кожної сфери існують свої нюанси.

Цікавою нішею для дослідження стало корпоративне програмне забезпечення для малого і середнього бізнесу. На даний момент існує багато програмних продуктів для комерційних цілей, проте вони здебільшого розраховані на великі корпорації і підприємства. Тому використання їх для менших фірм просто не доцільно, або дуже складно із фінансової сторони (витрати на підтримку передують вигоді). Як відомо, на ринку до цих пір зберігається тенденція на попит на корпоративне програмне забезпечення, яке б відповідало вимогам малого і середнього бізнесу, і в той же час було практично придатним для використання у великих корпоративних цілях

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ РИНКУ І ПОТОЧНИХ РІШЕНЬ

На даний момент існує досить багато готових рішень корпоративних порталів. Вони можуть забезпечувати підприємства всіма необхідними функціями і додатками, починаючи від системи обліку працівників, і завершуючи системою аналітики і збору даних, все це залежить від потреб ринку і певної компанії. Проте майже всі вони здебільшого призначені для великих компаній, і невеличкі компанії або повинні витратити величезні гроші на покупку ліцензій, або ж не користуватися усіма перевагами корпоративного порталу. Тому було проведено загальний огляд продуктів і виділено основні переваги і недоліки, також виділено поточну використовувану ліцензію розповсюдження ПЗ.

1.1 Переваги та недоліки поточних рішень

Зробивши аналіз даної сфери, можна виділити декілька основних аспектів, які будуть використані для розробки подальшого програмного продукту. Левова частка програмного забезпечення корпоративних порталів розроблено згідно стандартів[?]. Всі додатки і аплікації можуть без проблем взаємодіяти між собою. Проте великою їх нестачею для малої сфери бізнесу є закритість програмного коду і величезна вартість ліцензій. Тому невеличкі компанії (до 100 людей) просто не мають змоги собі дозволити таку «розкіш».

1.1.1 Ліцензіювання і відкритість API

Було проведено аналіз поточних продуктів і їх ліцензій, і виявлено, що майже 90% використовує пропріетарні рішення. Більш детально розглянуто у таблиці 1.1:

Табл. 1.1 – Список корпоративних порталів і використовувана ліцензія

Назва продукту	Технологія	Ліцензія
Jetspeed	Java EE	Apache License v2.0
ATG Portal	Java EE	Proprietary

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

Назва продукту	Технологія	Ліцензія
Backbase Portal Software	Java EE, .NET	Proprietary
Broadvision Portal	Java EE	Proprietary
Bluenog ICE	Java EE	Proprietary
enPortal	Java EE	Proprietary
CommunityManager.NET	.NET	Proprietary
eXo Portal	Java EE	Affero General Public License
eXo Platform	Java EE	Proprietary
GateIn Portal	Java EE	LGPL
Hippo CMS	Java EE	Open Source and Proprietary Licenses
WebSphere Portal	Java EE	Proprietary
TeamPortal	Java EE	Proprietary
JBoss Enterprise Portal	Java EE	LGPL
IntraNet	ASP.NET	Proprietary
Liferay Portal	Java EE	Proprietary Licenses
TeamWox Groupware	C++	Proprietary
SharePoint Server	ASP.NET	Proprietary
Vignette Portal 8.0	Java EE	Proprietary
Oracle WebCenter Suite 11g	Java EE	Proprietary
Oracle WebLogic Portal 10g	Java EE	Proprietary
Oracle WebCenter Interaction 10g	ASP.NET	Proprietary
Oracle IAS Portal 10g	Java EE	Proprietary
Regroup	Ruby	Proprietary
ACUBE Portal 5.0	Java EE	Proprietary
SAP NetWeaver 7.0	Java EE	Proprietary
SORCE V9	ASP.NET	Proprietary
Sun Java System Portal Server	Java EE	Open Source, licensing & support plans
Sun GlassFish Web Space Server	Java EE	Open Source, licensing & support plans
tmsEKP 1.52	Java EE	Proprietary
PortalBuilder 5.2	Java EE	Proprietary
ProPortal 4.0	Java EE	Proprietary
Intrexx	Java EE	Proprietary
uPortal	Java EE	Apache License v2.0

Зм.	Арк.	№ докум.	Підпис	Дата

ДП.ПЗ 04.00.00.000 ПЗ

Архив

5

Зробивши певний аналіз, можна дійти до висновку, якби невеликим компаніям давати можливість використовувати продукт на підставах вільного розповсюдження коду, то вони б охотніше пробували, і з часом інвестували в нього гроші, або просто «купляти» підтримку. Тобто використати одну із відкритих ліцензій типу GNU Public License.

Економічна вигода продукту буде базуватися на корпоративній платній підтримці, типу все починаючи від підбору серверів – до налаштування і підтримки продукту. Зате розробники зможуть у загальній репозиторії додавати свої зміни та виправляти помилки, що значною мірою пришвидшить процес розробки. Основна стратегія розробки буде націлена на швидкий вихід на ринок і пошук потенційних клієнтів. Також велика увага буде прикута для ринку пост радянських республік, адже на даний момент ринок бізнесу стрімко розвивається, тобто попит є, а пропозиція не повній мірі відповідає потребі. Портали які розробляються переважно націлені на Європейський та Американський ринок, а також Азію. Тому базуючись на цьому було виділено такі основні вимоги, як врахування нашого законодавства (для прикладу по працевлаштуванню працівників, ведення документації, конвертації валют і тому подібне) та локалізацію сервісу. Тим більше підтримка користувачів буде набагато легше і ефективніше всередині країни, ніж з-за кордону, що дасть нам перевагу над іншими існуючими продуктами.

Також велику увагу буде приділено відкритості API для взаємодії із вже існуючими додатками. Адже існуючі рішення в основному базуються на закритих протоколах, чим саме змушують користувачів прив'язуватися до їхньої системи і залежати від них

Інтерфейс і система всіх сучасних продуктів дуже «важка» і мультифункціональна, що потребує значних ресурсів як у користувачів так і на стороні сервера. Цей аспект також буде максимально спрощений, що в свою чергу дозволить виділитися продукту на ринку малого бізнесу.

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		6

1.2 Технології розробки

Базуючись на поточних стандартах [?] та використовуваних базових технологій (таблиця 1.1) було прийнято рішення впровадити розробку на базі Java. Адже саме Sun (зараз Oracle) «диктує» моду на ринку стандартизації портлетів, тому буде просто пристосуватися до поточних рішень.

Звичайно ж буде використано всі переваги Java EE. Для гнучкої і швидкої розробки буде застосовано Spring Framework із ORM обгорткою Hibernate поверх бази даних MySQL. Для фронтенд логіки UI в основному буде використовуватися jQuery фреймворк. Пошук забезпечить Apache Lucene. Сервер бекенду буде працювати на Apache Tomcat.

На даний момент не планується стандартизація щодо портлетів, просто в майбутньому можливо буде виділено цей пункт для реалізації в системі.

1.3 Технічний огляд продукту

Розглянемо більш детально пункт про наявність сучасних засобів ведення бізнесу. Кожна компанія, завжди стикається із проблемою ведення обліку працівників, ведення обліку фінансів, спільної роботи над документами та іншим. Також є величезна і невід’ємна потреба у спільному доступі до документів, до корпоративного календаря, до блогу користувачів, до електронних таблиць та інформаційної дошки.

Портал підприємства (також відомий як enterprise information portal (EIP) або корпоративний портал) є основою для інтеграції інформації, людей і процесів в рамках організації. Це дає змогу забезпечити єдину точку доступу, часто у вигляді веб-інтерфейсу і призначеної для агрегування та персоналізації інформації за допомогою конкретних програмних додатків. Однією відмінною рисою корпоративних порталів є децентралізоване внесення контенту та управління, яка зберігається на віддаленому сервері та постійно оновлюється.

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		7

1.4 Історичний огляд корпоративної сфери

В середині 1990-х років появилися громадські такі веб-портали як AltaVista, AOL, Excite і Yahoo!. Вони забезпечували користувачів певним набором функцій (наприклад новини, електронна пошта, погода, котирування акцій і пошук), які часто були представлені у вигляді автономного порталу. Незабаром підприємства усіх типів і форм почали бачити необхідність аналогічного функціоналу для їх різноманітних потреб, проте із єдиною точкою доступу.

До кінця 1990-х років, виробники програмного забезпечення почали розробляти веб-портали для різних підприємств. Ці програмні пакети були розроблені таким чином, щоб підприємства могли легко розгортати свої власні налаштування корпоративного порталу та доповнювати його своїми додатками. Перші постачальники комерційних веб порталів з'явилися в 1998 році, це були такі фірми як: Epicentric, Plumtree та Viador. Ці фірми були основними гравцями на ринку, проте ситуація змінилася в 2002 року, коли на ринок почали виходити постачальники серверних аплікацій, такі як BEA, IBM, Passageways, Oracle Corporation and Sun Microsystems. Підприємства можуть вибрати для своїх цілей декілька порталів, що базується на основі їх бізнес-структури та стратегічної спрямованості.

У 2003 році розробники Java-порталів випустили стандарт, відомий як JSR-168. Він повинен був визначити API для взаємодії між корпоративних порталів та портлетів. Постачальники програмного забезпечення почали розробляти JSR-168 сумісні портлети, які можуть бути розгорнуті на будь-якому JSR-168 сумісному корпоративному порталі. Другий ітераційний стандарт JSR-286 є остаточним на даний момент і випущений 12 червня 2008 року.

1.5 Портлети

Портлет – це змінний компонент інтерфейсу веб-порталу (елемент веб-сторінки), який можливо певним чином підключити до порталу. Портлет містить в собі фрагменти розмітки, які вбудовуються в сторінку порталу. Найча-

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		8

стіше сторінка portalу представляється у вигляді набору портлетів, які взаємодіють між собою. Таким чином, портлет (або сукупність портлетів) представляється у вигляді єдиного веб-додатку, розміщеного на порталі. Приклади портлетів можуть бути наступними: електронна пошта, повідомлення про погоду, фінансовий стан, останні новини і тому подібне. Завдяки існуючим стандартам розробники можуть створювати портлети, що легко вбудовуються в будь-який портал, який слідує стандартам і правилам.

Існує протокол WSRP, що забезпечує стандарт веб-сервісів, що дозволяє автоматично вбудовувати віддалено запуснені портлети з різних джерел. Специфікація Java-портлетів JSR168 дає можливість взаємодіяти між собою портлети з різних веб-порталів. Ця специфікація визначає безліч API для взаємодії контейнерів портлетів і дає різні адреси областей персоналізації, подання та безпеки. Існує безліч постачальників комерційних контейнерів портлетів. Як відомо лідирують у цій галузі IBM, Oracle, Vignette. Реалізації від цих постачальників мають додаткові розширення і налаштування, проте деякі із ним можуть бути не затверджені стандартами. Крім того, є портали з відкритим вихідним кодом, що підтримують JSR168, такі як корпоративний портал Apache Jetspeed-2 або eXo Portal.

1.5.1 *Apache Pluto*

Розглянемо на прикладі один з найбільш вдалої реалізації стандарту портлетів JSR168 – це Apache Pluto. Портлет працює всередині контейнера портлетів (Pluto). Цей контейнер містить портлет з необхідним середовищем для подальшого виконання. Контейнер портлетів керує життєвим циклом всіх вікон portalу та надає інтерфейси для портлетів, котрі викликаються всередині нього. Контейнер також запускає методи на виконання із доступних цільових користувацьких сторінок, і взаємодіє із сторінками portalу. Принцип роботи і архітектурні компоненти аплікації Pluto 2.0. зображено на рисунку 1.1.

В даному випадку, Pluto вбудований безпосередньо в корпоративний портал. Потім через перехресний запит (через веб-додатки) відбувається відправлен-

					ДП.ПЗ 04.00.00.000 ПЗ	Архив
Зм.	Арк.	№ докум.	Підпис	Дата		9

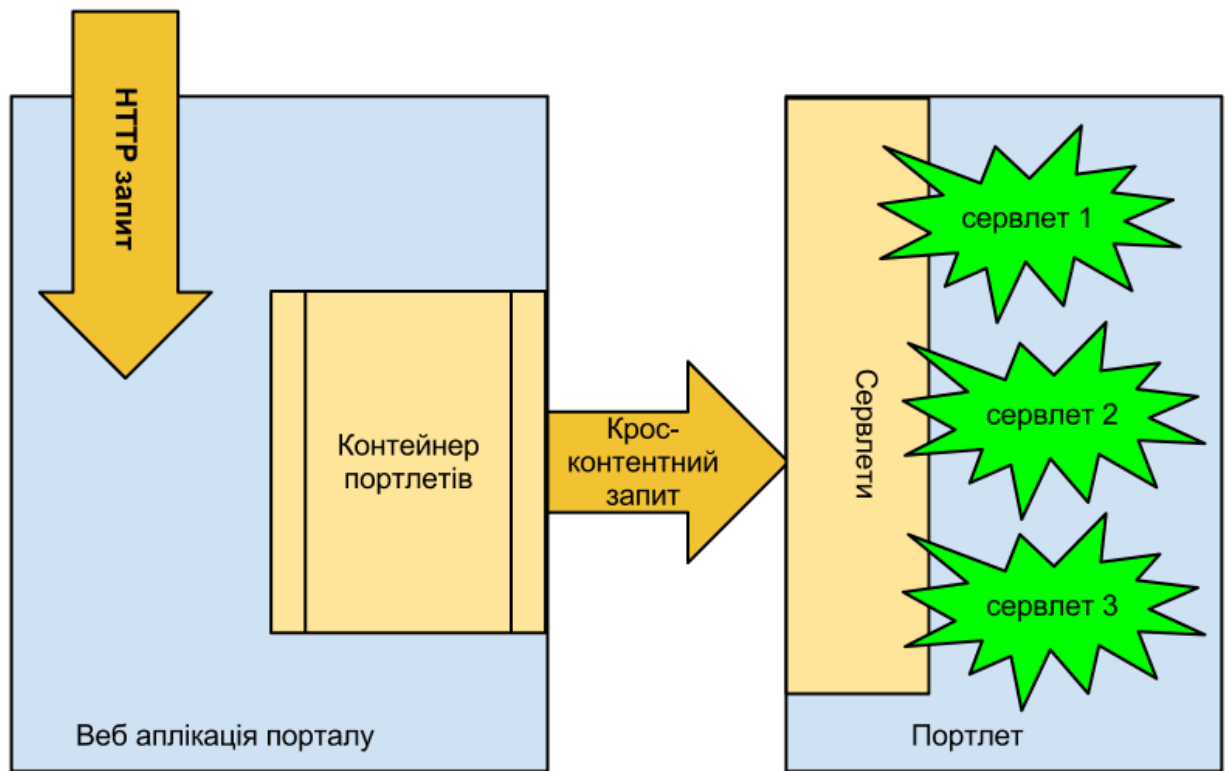


Рисунок 1.1 – Принцип роботи Pluto

ня запиту для відображення вмісту портлету, який як правило знаходяться в різних додатках на порталі і в контейнерах.

1.6 Система керування вмістом

Система управління контентом (content management system - CMS) дозволяє публікувати, редагувати і змінювати вміст веб-сторінок, а також обслуговувати портал з центральної сторінки. При цьому надається набір процедур, що використовуються для управління робочим процесом у середовищі для спільної роботи. Вони можуть бути ручні або комп'ютеризовані (в автоматичному режимі).

1.6.1 Головні функції CMS

До основних функцій можна віднести наступні пункти:

- а) можливість великій кількості людей ділитися інформацією і робити свій вклад в розвиток порталу;

- б) контроль доступу до даних на основі ролей користувачів (наприклад визначити роль, яка має тільки права на перегляд інформації, або ж редагування, публікацію тощо);
- в) пошук і поширення інформації між користувачами;
- г) зменшення дублікацій на вході;
- д) спрощене керування корпоративними додатками;
- е) відносно легка комунікація між користувачами.

1.6.2 Типи даних та їх використанням

У CMS дані можуть бути представлені як правило у будь-якій формі: документи, відео, тексти, фотографії, номери телефонів, наукові дані і тому подібне. CMS часто використовуються для зберігання, управління, перегляду і публікації документів. Також досить поширене використання в якості центрального сховища у зв'язці із централізованою системою контролю версій, що є однією із переваг CMS.

1.6.3 Управління корпоративною інформацією

Enterprise Content Management (ECM) - управління інформаційними ресурсами підприємства або управління корпоративною інформацією. В даному контексті інформація (контент) передбачається як слабо структурована одиниця - це можуть бути файли різних форматів, електронні документи з різними наборами полів і т. п. За визначенням ECM - це стратегічна інфраструктура і технічна архітектура для підтримки єдиного життєвого циклу неструктурованої інформації різних типів і форматів. ECM-системи складаються з додатків, які можуть взаємодіяти між собою, а також використовуватися і продаватися самостійно.

Всі сучасні ECM-системи визначають такі ключові компоненти:

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

- а) управління документами — довгострокове архівування, автоматизація політик зберігання та відповідності нормам регулюючих органів, забезпечення відповідності законодавчим та галузевим нормам;
- б) управління веб-контентом (WCM) — автоматизація ролі веб-майстра, управління динамічним контентом і взаємодією між користувачами;
- в) управління мультимедіаконтентом (DAM) — управління графічними, відео та аудіофайлами, різними маркетинговими матеріалами, наприклад, флеш-банерами, рекламними роликами;
- г) управління знаннями (Knowledge Management) — підтримка систем для накопичення та доставки релевантної для бізнесу інформації;
- д) документо-орієнтоване взаємодія (співробітництво) — спільне використання документів користувачами та підтримка проектних команд.

1.7 Система управління документами

Система управління документами (DMS - Document management system) - комп'ютерна система (або набір комп'ютерних програм), що використовується для відстеження та зберігання електронних документів і / або образів (зображень та інших артефактів) паперових документів. Дане поняття тісно пов'язане з концепцією Content Management System (система керування вмістом) і зазвичай розглядається як компонент Enterprise Content Management System (CMS рівня підприємства). У загальному випадку системи управління документами (DMS) надають можливість зберігання, ведення контролю версій, позначення метаданими і безпеку по відношенню до документів, а також індексування і розвинені можливості пошуку документів.

1.7.1 Метадані

Метадані зазвичай зберігаються для кожного документа. Метадані, наприклад, можуть включати дату занесення документа в сховище і код користувача,

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		12

котрий виконав зміни до файлу. Система управління документами також може витягувати метадані з документа автоматично або запитувати їх у користувача. Деякі системи надають сервіс оптичного розпізнавання тексту відсканованих документів, або можливість витягувати текст з електронних документів. Використовуючи опрацьований текст система дозволяє здійснювати пошук документа за ключовими словами всередині самого документа.

1.7.2 Інтеграція

Багато систем управління документами намагаються інтегрувати функцію управління документами безпосередньо в різні додатки, дозволяючи користувачеві отримувати документ відразу зі сховища системи управління документами, робити які-небудь модифікації, і зберігати його назад в сховище в якості нової версії, і все це проробляти в одному додатку, не виходячи з нього. Дана інтеграція в основному доступна для офісних пакетів і поштових клієнтів або для програмного забезпечення, призначеного для групової або колективної роботи. Інтеграція зазвичай має на увазі використання таких відкритих стандартів як: ODMA, LDAP, WebDAV і SOAP.

1.7.3 Захоплення тексту

Під захопленням тексту мається на увазі переведення паперових документів в цифровий варіант за сканерів та МФУ. Також часто використовується програмне забезпечення для оптичного розпізнавання тексту, щоб конвертувати цифрові зображення в текст.

1.7.4 Індексування

Індексування надає можливість класифікувати документи за допомогою метаданих і індексування словникового тексту, який було витягнутого з документа. Індексція існує для підтримки розвинених можливостей пошуку документів.

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						13
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Одна з головних умов швидкого та якісного пошуку - це створення індексу документа.

1.7.5 Сховище

Основне призначення це для зберігання електронних версій документів. Сховище документів також включає в себе і керування тими ж документами, котрі воно зберігає. Також сховище забезпечує міграцію з одного носія на інший і забезпечує цілісність даних. Сховище документів може бути як файлове, так і сховище у вигляді СУБД (бази даних). У свою чергу, сховище документів в СУБД може бути як в одній базі даних, так і в окремо розподілених базах даних.

1.8 Програмне забезпечення для спільної роботи

Програмне забезпечення для спільної роботи (англ. collaborative software, groupware, workgroup support systems, group support systems) - програмне забезпечення створене з метою підтримки взаємодії між людьми, котрі спільно працюють над вирішенням деяких спільних завдань.

1.8.1 Огляд ПЗ для спільної роботи

Програмне забезпечення для спільної роботи – це область, яка в значній мірі перекривається з областю CSCW (англ. computer-supported cooperative work (CSCW)). Часто вважається що ці області еквівалентні, хоча з іншого боку програмне забезпечення для спільної роботи є підчастиною CSCW. Сюди відносяться такі системи як: електронна пошта, календарі, текстовий чат, вікі сторінки, корпоративні закладки, блог. Оскільки ПО спільної роботи відноситься до технологічних елементів CSCW, системи спільної роботи стають корисним аналітичним інструментом у вивченні поведінкових і організаційних параметрів, пов'язаних з більш широкою сферою CSCW.

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

1.8.2 Види взаємодії

В літературі можна зустріти кілька різних визначень спільної роботи (англ. - collaboration) в застосуванні до інформаційних технологій. Деякі з них виправдані, інші ж настільки великі, що починають втрачати будь-який сенс. Для того щоб бути впевненим що обрані технології підходять для конкретних потреб, необхідно розуміти відмінності в способах взаємодії людей один з одним. Є три основні шляхи, по яких здійснюється взаємодія між людьми:

- а) діалог;
- б) здійснення угоди;
- в) співробітництво.

Діалог - це обмін інформацією між одним або кількома учасниками, основна мета якого полягає у з'ясуванні їх позицій і встановлення взаємин. Відбувається вільний обмін інформацією без будь-яких обмежень. Для підтримання діалогу цілком підходять звичайні комунікаційні технології, такі як телефон, миттєві повідомлення та електронна пошта.

Укладення угоди передбачає обмін якимись сутностями, і ця процедура зазвичай проводиться за добре певними правилами і передбачає зміну відносин між учасниками. Наприклад, один з учасників угоди обмінює гроші на товари і стає покупцем. Новий статус учасників операції та обмінюваних сутностей потрібно зберегти в будь-якому надійному сховищі. Такі операції добре обслуговуються системами управління транзакціями.

Співпраця полягає в тому, що його учасники обмінюються якимись загальними сутностями (на противагу угоді, коли предмет обміну належить лише одному учаснику). Як приклад можна привести просування нової ідеї, створення нової конструкції, досягнення спільних цілей. При цьому самі сутності досить розпливчасті і невизначені. Таким чином, технології для забезпечення спільної роботи теж повинні бути достатньо гнучкими. Вони повинні включати в себе управління

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		15

документами, кошти для ведення обговорень з можливістю сортування за темами, можливість відновити історію внесених змін та багато іншого.

1.8.3 Рівні взаємодії

Рівні взаємодії можна поділити на три категорії по рівню забезпечення взаємодії: засоби зв'язку, засоби для організації конференцій та засоби управління.

Електронні засоби зв'язку використовуються для пересилання повідомлень, файлів, даних чи документів між людьми і таким чином дають можливість для обміну інформацією:

- а) електронна пошта;
- б) факс;
- в) голосова пошта;
- г) веб-публікації.

Електронні конференції також дають змогу для обміну інформацією, проте в інтерактивній формі це є:

- а) телефонні конференції;
- б) відео і аудіо конференції;
- в) інтернет форуми;
- г) чати.

Засоби управління діяльністю групи:

- а) електронні календарі (створення щоденників, системи автоматичного нагадування);
- б) системи управління проектами (складання розкладу робіт, відслідковування їх виконання);
- в) управління документообігом;

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

- г) бази знань - збір, сортування, зберігання і організація доступу до різних форм інформації.

1.9 Інтранет

Інтранет (англ. Intranet, також вживається термін інтрамережа) - на відміну від мережі Інтернет, це внутрішня приватна мережа організації. Як правило, Інтранет - це Інтернет в «мініатюрі», який побудований на використанні протоколу IP для обміну і спільного використання деякої частини інформації всередині певної організації. Це можуть бути списки співробітників, списки телефонів партнерів і замовників. Найчастіше під цим терміном мають на увазі тільки видиму частину Інтранет - внутрішній веб-сайт організації. Заснований на базових протоколах HTTP і HTTPS і організований за принципом клієнт-сервер, інтранет-сайт доступний з будь-якого комп'ютера через браузер.

Таким чином, Інтранет - це «приватний» Інтернет, обмежений віртуальним простором окремо взятої організації. Intranet допускає використання публічних каналів зв'язку, що входять в Інтернет, (VPN), але при цьому забезпечується захист переданих даних і мають набір заходів щодо припинення проникнення ззовні на корпоративні вузли.

Програми в Intranet засновані на застосуванні Інтернет-технологій і особливо Web-технологій: гіпертекст у форматі HTML, протокол передачі гіпертексту HTTP і інтерфейс серверних додатків CGI. Складовими частинами Intranet є Web-сервери для статичної або динамічної публікації інформації і браузери для перегляду й інтерпретації гіпертексту.

1.9.1 Особливості, переваги та недоліки Інтранет

Інтранет побудований на базі тих же понять і технологій, які використовуються для Інтернету, такі як архітектура клієнт-сервер і стек протоколів Інтернету (TCP / IP). В Інтранет зустрічаються все з відомих інтернет-протоколів, наприклад, протоколи HTTP (веб-служби), SMTP (електронна пошта) і FTP (передача

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		17

файлів). Інтернет-технології часто використовуються для забезпечення сучасними інтерфейсами функції інформаційних систем, які розміщують корпоративні дані.

Інтранет можна представити як приватну версію Інтернету, або як приватне розширення Інтернету, обмеженого організацією за допомогою брандмауера.

Перші інтранет-веб-сайти і домашні сторінки почали з'являтися в організаціях у 1990-1991 роках. Проте за неофіційними даними, термін Інтранет вперше почав використовуватися в 1992 році в таких закладах, як університети і корпорації, що працюють у технічній сфері.

Інтранет також протиставляють Екстранет, доступ до Інтранету надано тільки службовцям організації, в той час як до Екстранет можуть отримати доступ клієнти, постачальники, або інші затверджені керівництвом особи. В Екстранет-технології крім приватної мережі, користувачі мають доступ до Інтернет ресурсів, але при цьому здійснюються спеціальні заходи для безпечного доступу, авторизації, і аутентифікації.

Інтранет компанії не обов'язково повинен забезпечувати доступ до Інтернету. Коли такий доступ все ж забезпечується, зазвичай це відбувається через мережевий шлюз з брандмауером, захищаючи Інтранет від несанкціонованого зовнішнього доступу. Мережевий шлюз часто також здійснює аутентифікацію користувачів, шифрування даних, і часто - можливість з'єднання по віртуальній приватній мережі (VPN) що знаходяться за межами підприємства.

Переваги використання Інтранет:

- а) висока продуктивність при спільній роботі над якимись загальними проектами;
- б) легкий доступ персоналу до даних;
- в) гнучкий рівень взаємодії: можна міняти бізнес-схеми взаємодії як по вертикалі, так і по горизонталі;
- г) миттєва публікація даних на ресурсах Інтранет дозволяє специфічні кор-

поративні знання завжди підтримувати у формі і легко отримувати звідусіль в компанії, використовуючи технології Мережі та гіпермедіа;

- д) дозволяє проводити в життя загальну корпоративну культуру і використовувати гнучкість і універсальність сучасних інформаційних технологій для управління корпоративними роботами.

Переваги веб-сайту в Інтранет перед клієнтськими програмами архітектури клієнт-сервер:

- а) Не потрібно інсталяція програми-клієнта на комп'ютерах користувачів (як неї використовується браузер).
- б) Відповідно, при змінах функціональності корпоративної інформаційної системи оновлення клієнтського ПЗ також не потрібно.
- в) Скорочення тимчасових витрат на рутинних операціях по вводу різних даних, завдяки використанню веб-форм замість обміну даними по електронній пошті
- г) Крос-платформна сумісність.

Основні недоліки Інтранет:

- а) мережа може бути зламана і використана в хакерських цілях;
- б) неперевірена або неточна інформація, опублікована в Інтранет, призводить до плутанини і непорозумінь;
- в) легкий доступ до корпоративних даних може спровокувати їх витік до конкурентів через несумлінного працівника;
- г) працездатність і гнучкість Інтранет вимагають значних накладних витрат на розробку і адміністрування.

1.10 Корпоративна Wiki

Корпоративна вікі — це програмне забезпечення яке призначене для використання в корпоративній сфері і служить особливим чином для підвищення

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		19

внутрішнього обміну знаннями, з великим акцентом на такі функції, як контроль доступу, інтеграція з іншими програмними продуктами та управління документами.

В організаціях вікі може або додати або замінити централізовану систему керування контентом. Її децентралізований характер дозволяє швидкому поширенню необхідної інформації в межах організації. Вікі являється швидшим організаційним продуктом ніж централізований репозиторій знань. Вікі може використовуватися для управління проектами, взаємодією з клієнтами, планування ресурсів підприємства а також інші види управління даними.

Особливості вікі для корпорації включають в себе такі основні аспекти як:

- а) швидкий і простий доступ для створення сторінок, які містять посилання на інші корпоративні системи;
- б) дозволяє розвантажити електронну пошту за рахунок зберігання всієї необхідної інформації із можливістю спільного доступу людьми які є на даному проекті.
- в) гнучка організація інформації;
- г) швидкий і розширений пошук.

1.11 Онлайн офіс

Онлайн офіс — це набір веб-сервісів у формі програмного забезпечення яке подану кінцевому користувачеві як послуга. Набір наданих веб-служб зазвичай включає всі основні можливості традиційних офісних пакетів, такі як текстовий редактор, електронні таблиці, додаток для створення презентацій, органайзер справ і навіть аналоги СУБД. Онлайн офіс може бути доступний з будь-якого комп'ютера, у якого є доступ в Інтернет, незалежно від того, яку операційну систему користувач використовує. Це дозволяє людям працювати разом по всьому світу і в будь-який час, що веде до створення міжнародних віртуальних команд для спільної роботи над проектами.

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

1.12 Корпоративний блог

Корпоративний блог — це блог, що видається організацією і використовується як для зв'язків з громадськістю, так і для внутрішньої організації. Або повністю підконтрольний організації, координований і наповнюється нею контентом, але формально з нею не пов'язаний.

1.12.1 Внутрішньокорпоративний блог

Внутрішній корпоративний блог — це важливий засіб комунікації, особливо у великих компаніях. Можна навести деякі явні переваги:

- а) блог допомагає поліпшити взаємодію співробітників, надає можливості для навчання. Він добре підходить для запуску нових проектів, для роботи в неоднорідних, великих колективах;
- б) блог допомагає виявити різні погляди на будь-яке питання. Відкритість для публікації постів і коментарів — хороша можливість висловитися всім членам колективу;
- в) шляхом дискусій на задану тему блог допомагає знайти компроміс при наявності різних точок зору. Для керівників блог — можливість налагодити взаємодію з співробітниками;
- г) блог — це своєрідна «історія фірми», архів ідей і обговорень.
- д) найчастіше кожен співробітник може залишити коментар до будь-якого посту. Коло авторів блогу визначається політикою компанії, часто написати пост може будь-який співробітник.

Блог має певні переваги перед такими внутрішньокорпоративними комунікаціями, як, наприклад, листування по електронній пошті, зокрема:

- а) коли листів стає занадто багато, це ускладнює спілкування;

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

- б) не всі співробітники вміють правильно архівувати листи, в результаті чого вони не зможуть згодом знайти необхідну інформацію.

Внутрішній блог — альтернатива чи доповнення до корпоративних зборів, нарад. Співробітники великих компаній часто не мають можливість проводити наради (наприклад, через велику відстань між філіями або зайнятості).

1.12.2 Публічний блог

Одна з основних цілей компаній — це налагодження комунікацій з клієнтами (як поточними, так і потенційними). Завдяки оперативності публікації постів і можливості коментування публічний корпоративний блог дуже важливий для досягнення цієї мети. Блоги є цінним доповненням до корпоративного сайту, так як в них може бути представлена альтернативна точка зору на те чи інше питання, ті чи інші продукти компанії можуть бути описані більш простою і доступною мовою.

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						22
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

2 АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ПРОЦЕСУ СТВОРЕННЯ КОРПОРАТИВНОЇ СИСТЕМИ

2.1 Проектування системи

На початку розробки будь-якого програмного продукту слід значну увагу приділити проектуванню системи, адже саме від цього буде залежати легкість і правильність подальшої розробки системи, її підтримка і удосконалення. Саме тому проектування визначає основну складову програмного забезпечення. Необхідні пункти для успішного запуску продукту:

- а) побудова UML діаграми класів;
- б) побудова діаграми відношень між об'єктами;
- в) проектування бази даних;
- г) створення макетів майбутнього інтерфейсу;
- д) чітке розмежування модулів системи і їх взаємодія і тому подібне.

Як було згадано вище, перш за все слід розробити діаграму класів, показати всі взаємовідношення між об'єктами, їх роль у системі та загальну взаємодію.

2.1.1 Модель архітектури системи

Вся робота системи організована за принципом MVC шаблону.

Модель-вид-контролер – архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Як показана на рисунку 2.1 цей шаблон поділяє систему на три частини: модель даних, вигляд даних та керування. Застосовується для відокремлення даних (модель) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

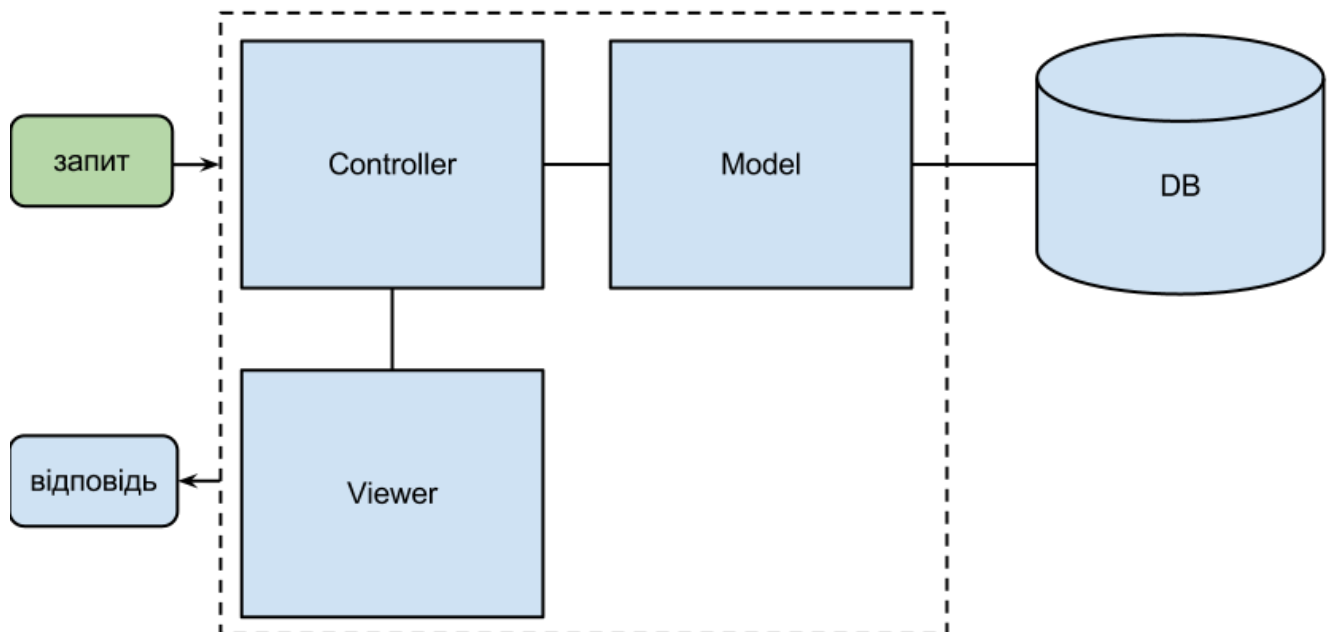


Рисунок 2.1 – Архітектура MVC

Головна мета використання даного шаблону – гнучкий дизайн програмно-го забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонент програми. Крім того використання цього шаблону у великих системах призводить до певної впорядкованості їх структури і робить їх зрозумілішими завдяки зменшенню складності.

Як згадувалося раніше, архітектурний шаблон Модель Вид Контролер (MVC) поділяє програму на три частини. У тріаді до обов’язків компоненту Модель (Model) входить зберігання даних і забезпечення інтерфейсу до них. Вигляд (View) відповідальний за представлення цих даних користувачеві. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача, і повідомляє про зміни компоненту Модель. Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами.

MVC поділяє цю частину системи на три самостійні частини: введення даних, компонент обробки даних і виведення інформації. Модель, як вже було відмічено, інкапсулює ядро даних і основний функціонал з їх обробки. Також ком-

понент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вигляд може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація. У функції Контролера входить моніторинг за подіями, що виникають в результаті дій користувача (зміна положення курсора миші, натиснення кнопки або введення даних в текстове поле). Зареєстровані події транслюються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.

2.2 Проектування бази даних

База даних базується на СКБД MySQL - відкритій системі. Взаємозв'язок із користувачами відбувається через Модель системи (див. рис. 2.1). В базі містяться всі необхідні дані. Паролі користувачів зберігаються у шифрованому вигляді.

Реляційна система керування базами даних (РСКБД; інакше Система керування реляційними базами даних, СКРБД) — СКБД, що керує реляційними базами даних.

Поняття реляційний (англ. relation – відношення) пов'язане з розробками відомого англійського спеціаліста в області систем баз даних Едгара Кодда (Edgar Codd).

Ця модель характеризується простотою структури даних, зручним для користувача табличним представленням і можливістю використання формального апарату алгебри відношень і реляційного обчислення для обробки даних.

Реляційна модель орієнтована на організацію у вигляді двовимірних таблиць. Кожна реляційна таблиця являє собою двовимірний масив і має такі властивості:

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		25

- а) кожний елемент таблиці - один елемент даних;
- б) всі комірки в стовпці таблиці однорідні, тобто всі елементи в стовпці мають однаковий тип;
- в) кожний стовпець має унікальне ім'я;
- г) однакові рядки в таблиці відсутні;
- д) порядок наступності рядків і стовпців може бути довільним.

Базовими поняттями реляційних СКБД є:

- а) атрибут;
- б) відношення;
- в) кортеж.

2.2.1 InnoDB механізм

InnoDB це потужний механізм (рушій) для зберігання даних, розроблений фінською компанією Innobase Oy, яка була придбана в 2006 році концерном Oracle Corporation.

Поширюється за ліцензією GNU General Public License. Є у всіх нових версіях MySQL, і, починаючи з версії 5.5 для MySQL механізм за замовчуванням.

Застосування InnoDB дозволяє використання базою даних таких функцій, як транзакції, зовнішні ключі. Він також сумісний з ACID.

У цьому рушії є два способи для зберігання даних: файл або група файлів, загальних для всіх баз даних і таблиць, або один файл даних для кожної таблиці. Інші важливі особливості InnoDB: блокування на рівні рядків, можливість стиснення даних, і MVCC.

2.2.2 Відношення в таблицях

Головною одиницею бази даних є таблиця яка відповідає за дані користувачів, адже саме від неї залежать більшість таблиць. Для прикладу це повідомлення,

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

чи завдання.

Для зв'язку між таблицями використовуються зовнішні ключі. Зовнішній ключ це – атрибут (набір атрибутів) в деякому відношенні R, який відповідає первинному ключу іншого відношення або того ж таки відношення R.

Ці взаємозв'язки представляються у вигляді відношень. Розділяють три види відношень:

- а) Багато-до-багато (n:m)
- б) Один-до-багато (1:m)
- в) Один-до-одного (1:1)

Багато-до-багатьох SQL відносин використовується, коли деяка невизначена кількість рядків (n) в таблиці пов'язані з невизначеною кількістю рядків (m), які зберігаються в іншій таблиці. Це називається: (m:m) відносини, тому що (n) рядків у першій таблиці, відносяться до (m) рядків в іншій.

Потрібно бути впевненим, що використовується кількість рядків строго більше ніж одиниця, тому що у випадку із одиницею слід використовувати відношення один-до-багатьох (1:n).

Розглянемо приклад, який зберігає країни в одну таблицю і мови в іншу таблицю. Є країни в світі, де більш ніж одна офіційний мова, і відповідно є випадки коли говорять одною мовою більш ніж в одній країні. Саме для цього призначене відношення багато-до-багатьох.

Приклад створення такої бази даних продемонстрований нище:

```
<c:choose>
<c:when test="{menu eq 'NOTE' }"><li class="active"><a class="
  nav-icon icon-note" href="/ccms/notes">Notes</a></li></c:when>
<c:otherwise><li><a class="nav-icon icon-note" href="/ccms/notes
  ">Notes</a></li></c:otherwise>
</c:choose>
```

Реалізація такої структури таблиць зображено на рисунку 2.2.

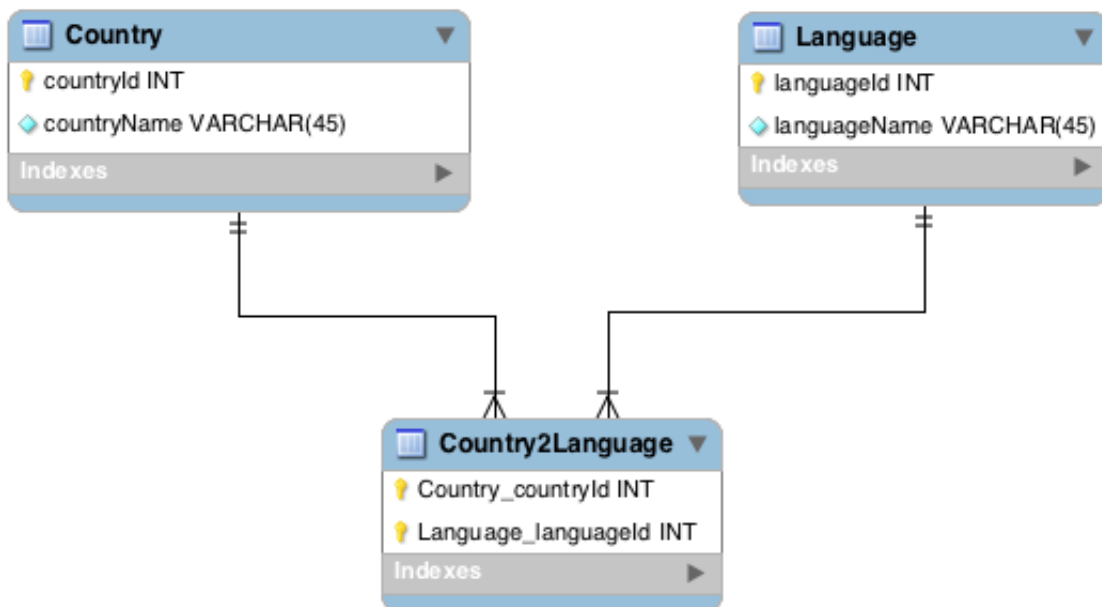


Рисунок 2.2 – Відношення багато-до-багато

Один-до-багато (1: n) відношення є дуже поширеним в SQL. Основна ідея полягає в тому, що кожен рядок зберігається в одній таблиці та пов'язаний з невизначеною кількістю рядків у іншій таблиці. Це може бути будь-яке число між 0 і (n) рядків.

Дуже хорошим прикладом є SQL база даних, яка зберігає замовлення в одній таблиці (перша частина) і порядок позицій в іншій (n-на сторона). Приклад такої структури таблиць зображено на рисунку 2.3.

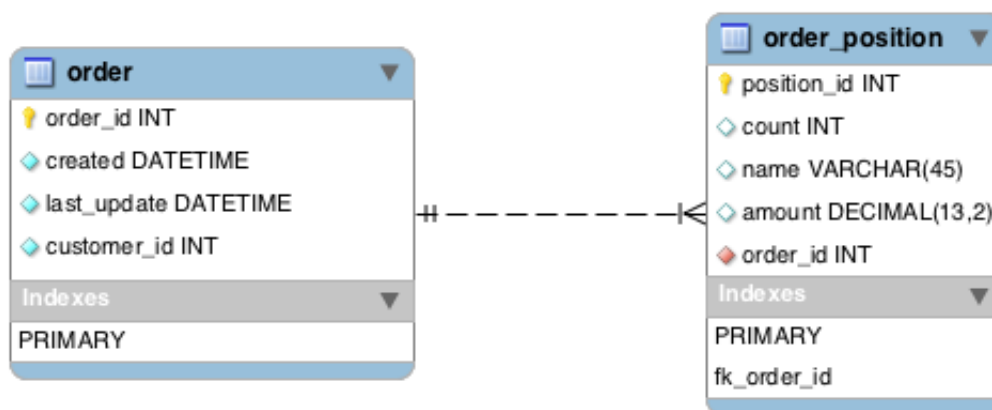


Рисунок 2.3 – Відношення один-до-багато

Реалізувати відношення один-до-багатьох дуже просто. Відносини зберігаються за допомогою спеціальної колонки в n-таблиці. Ця колонка містить один первинний ключ. Якщо присутні ще первинні ключі, то просто потрібно додати

ще декілька стовпців.

У наведеному вище прикладі, потрібно додати до основної колонки таблиці «order» (поле «order_id») позицію в таблицю «order_position». Для того щоб забезпечити цілісність даних, використовуються зовнішні ключі.

Основною перевагою один-до-багатьох є доступність нормалізації таблиць і не при цьому не потрібно нагромаджувати таблицю непотрібними даними.

Один-до-одного відношення в SQL використовують для того щоб розділити великі таблиці на менші при цьому без втрат продуктивності. В деяких випадках це краще ніж величезна EAV [?] таблиця.

Ідеальний варіант використання відносин 1:1 для не пов'язаних зв'язків, які поділяють основні ознаки, як для прикладу каталог записів. Приклад такої структури таблиць зображено на рисунку 2.4.

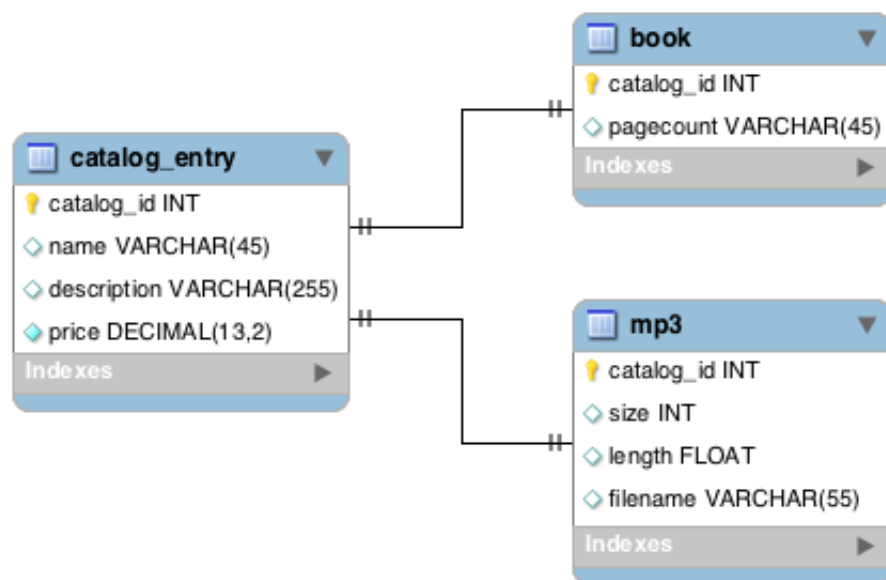


Рисунок 2.4 – Відношення один-до-одного

Ідея цієї концепції полягає в тому, що таблиця, яка містить основні атрибути, які є загальною для всіх суб'єктів. У першому каталозі загальні атрибути містять назва продукту, опис і ціну для прикладу. А самі атрибути, які є специфічними для певних видів продукції зберігаються в окремих таблицях. Зв'язок між цими таблицями зберігається за допомогою тих же первинний ключів в кожній таблиці.

Як ви можете бачити в наведеному вище прикладі, один-до-одного може бути реалізовано за допомогою простого використання тих же первинних ключів для таблиці.

Крім того, можливо використовувати різні первинні ключі і додавати відносини кожній колонці таблиці, але цей підхід має свої недоліки: SQL буде досить складним і заплутаним, і якщо використовувати зовнішні ключі, то це призведе до кругової залежності.

Первинний ключ – атрибут, або набір атрибутів, що однозначно ідентифікує кортеж даного відношення. Первинний ключ обов’язково унікальний, він єдиний і найголовніший із унікальних ключів. В реляційних базах даних первинний ключ задається обмеженням PRIMARY KEY. Для прикладу:

```
CREATE TABLE users(id INTEGER PRIMARY KEY AUTO_INCREMENT, name
CHAR(20), surname CHAR(40));
```

2.2.3 Архітектура бази даних

Як згадувалося вище майже все базується на відношенні до таблиці користувачів. Відношення повинні базуватися на основі моделі один-до-багатьох. Загальна модель бази даних зображена на рисунку 2.5.

Як показано на рисунку 2.5 – всі таблиці у базі даних взаємозв’язані. На таблицю користувачів посилаються таблиці документів, коментарів, блогів, календарів, команд, регіонів, типу робіт, вікі, завдань та повідомлень. Це в свою чергу дає змогу забезпечити відношення користувача до певної категорії. Для прикладу один користувач може мати декілька документів чи коментарів, звідси і слідує використання відношення один-до-багатьох.

Також допоміжні таблиці як категорія документів, кімнати, категорія вікі та категорія завдань мають свої відношення на головні таблиці (для прикладу одна категорія завдань може містити багато завдань).

					ДП.ПЗ 04.00.00.000 ПЗ	Архив
Зм.	Арк.	№ докум.	Підпис	Дата		30

2.3 Зовнішній макет сайту

Основою для користувача є зовнішній вигляд - UI (user interface) інтерфейс користувача. Адже кінцевому користувачу не завжди цікаво що відбувається всередині програми – тому розробляють макет майбутнього проекту, в нашому випадку – це макет веб інтерфейсу.

Макет інтерфейсу будується в звичайному графічному редакторі. Головною метою даної розробки є максимальне відображення всіх графічних властивостей майбутнього робочого веб-сайту. Макет майбутнього сайту, зокрема головної сторінки зображено на рисунку 2.6.

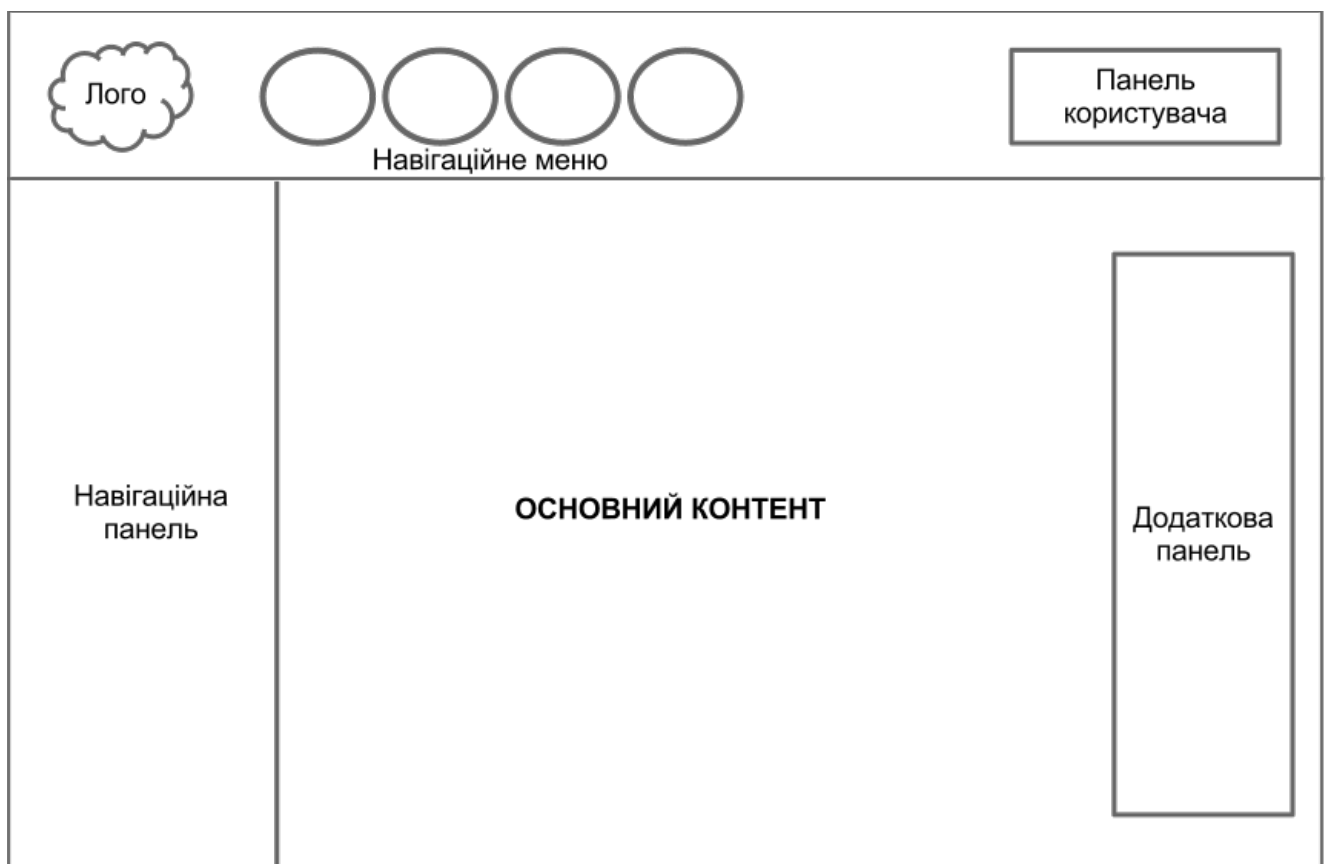


Рисунок 2.6 – Макет майбутнього сайту

З рисунку 2.6, де зображено макет сайту добре видно, що навігаційне меню розміщено зверху сайту. Дане меню повинно бути завжди доступне користувачу, тому воно завжди буде рухатися, тобто при будь якому положенні сторінки, не важливо чи внизу чи на верху, ця панель буде завжди доступна і рухати за користувачем. На ній варто розмістити швидкі навігаційні кнопки, такі як: швидке

створення завдання чи нотатки.

Зліва від всього розміщено навігаційну панель. Саме на цій панелі розміщуються всі доступні користувачеві на веб-проекті пункти навігації. В залежності від того де в даний момент перебуває користувач, слід підсвічувати поточне меню.

Панель користувача розміщується справа на навігаційній панелі. Тут повинна розмістися вся інформація для керування поточним користувачем: вихід із сайту, посилання на профіль користувача, приватні налаштування.

2.4 Структура проекту

Після побудови структури бази даних та загального макету сайту, слід приступити до розробки серверної частини – загальної структури сайту. Весь проект базується на роботі Java EE та поверх фреймворку Spring.

2.4.1 Алгоритм роботи надбудови Java EE – Spring Framework

Spring Framework працює наступним чином TODO TODO про всякі аспектно орієнтоване програмування і купа приколів для текстів.

2.4.2 Діаграма відношень

Діаграма відношень це – TODO Діаграма відношень користувача до вибірки із даних бази даних і їх формування зображені на рисунку 2.7

На рисунку 2.7 – діаграми відношень зображено весь процес трансферу даних – починаючи від запиту користувача, і завершуючи поверненням сформованих даних. Спочатку запит попадає на Asegi Security Controller.

Asegi – це TODO

Після проходження перевірки користувача (детально описано в розділі 2.5) дані від користувача попадають на контролер, звідки всі дані для запиту передаються на модель, яка відповідає за комутацію бази даних та контролера. Після

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		33

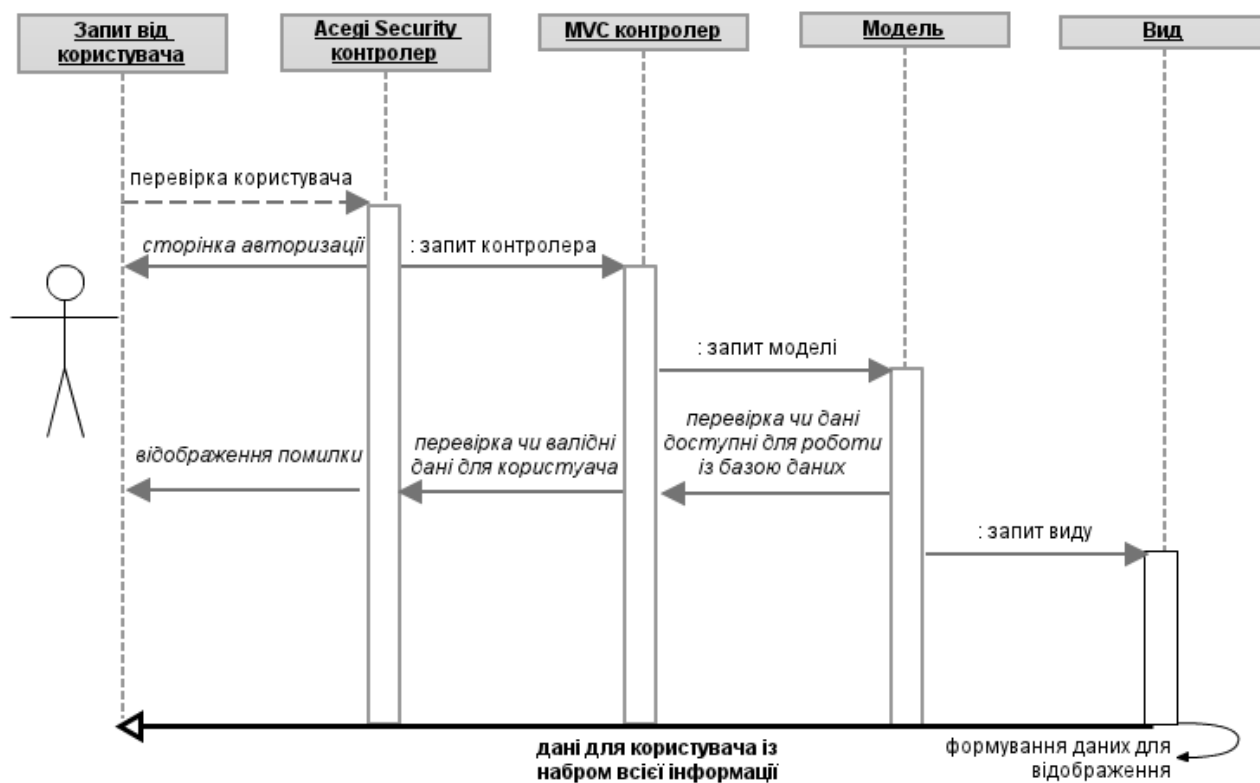


Рисунок 2.7 – Діаграма відношення запиту користувача до роботи сервера

успішних проходжень цих пунктів, на контролері відбувається формування вигляду майбутньої сторінки, яке будується на базі певного виду. Після цих всіх успішних маніпуляцій, сформовані дані повертаються користувачеві, тобто на веб інтерфейс. Якщо на якомусь етапі сталася помилка, чи невірна вибірка, то кожний етап зупиняється і повертається на попередній, аж до користувача із статусом помилки.

2.5 Авторизація і аутентифікація

Основою будь-якої корпоративної системи є можливість використання системи управління користувачами. Тому слід розробити повний цикл взаємодії користувачів. Сюди повинні бути включені наступні важливі аспекти:

- а) можливість авторизації користувачів;
- б) ролі користувачів;

Зм.	Арк.	№ докум.	Підпис	Дата

ДП.ПЗ 04.00.00.000 ПЗ

- в) зберігання даних у закодованому вигляді;
- г) чіткий розподіл прав користувачів;
- д) легка взаємодія між користувачами;
- е) можливість інтеграції із іншими сервісами системи.

Кожний працівник (він же користувач системи) повинний мати безперебійний доступ до свого профілю в будь-який час. Авторизація повинна бути реалізована інтуїтивно зрозуміло для кожного користувача і легко доступна. Управління користувачами буде реалізовано через адміністративну панель, доступ до якої будуть мати тільки користувачі певної групи.

2.5.1 Алгоритм авторизації користувачів

Загальний алгоритм авторизації полягає в перевірці даних користувача і повернення від серверу сформованих даних.

Сервер спочатку приймає надіслані від користувача логін та пароль. Логін передається на Acegi Security Controller і перекодовує пароль в шифр за допомогою алгоритму SHA-256. Адже зберігати дані не в шифрованому вигляді досить небезпечно, і у випадку, якщо зловмисник отримає їх, від цих даних не буде користі, оскільки вони шифровані і зворотнього шифру не існує. Шифрування на базі серверу відбувається через менеджер авторизації і налаштовується в конфігураційному файлі із посиланням на Bean із сервісом котрий відповідає за маніпуляцію даних користувача.

```
<authentication-manager alias="authenticationManager">
    <authentication-provider user-service-ref="
        UserServiceBean">
        <password-encoder hash="sha-256" />
    </authentication-provider>
</authentication-manager>

<beans:bean id="UserServiceBean" class="com.diploma.ccms.service.
    UserService" />
```

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		35

Слід виділити наступні групи користувачів:

- а) Адміністратори;
- б) Відділ кадрів;
- в) Користувач без особливих прав доступу.

Кожній групі повинний бути забезпечений доступ до певної категорії. Це можливо реалізувати за допомогою інтрацепторів. В разі доступу користувача до недозволеної категорії, буде повернена від сервера помилка.

```
<http auto-config="true" use-expressions="true">
<form-login login-processing-url="/resources/
    j_spring_security_check" login-page="/login" authentication-
    failure-url="/login?login_error=t" />
<logout logout-url="/resources/j_spring_security_logout" />
<!-- Configure these elements to secure URIs in your application
-->
<intercept-url pattern="/admin/**" access="hasRole('ROLE_ADMIN') "
    />
<intercept-url pattern="/login/**" access="permitAll" />
<intercept-url pattern="/resources/**" access="permitAll" />
<intercept-url pattern="/**" access="isAuthenticated()" />
</http>
```

2.6 Діаграма класів

Діаграма класів це – TODO з вікі

Взаємозв'язок всіх класів і інтерфейсів зображено на рисунку !!!

З рисунку видно що, ля, ля, ля, ля, ля, ля...

					ДП.ПЗ 04.00.00.000 ПЗ	Архив
Зм.	Арк.	№ докум.	Підпис	Дата		36

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЛЯ УПРАВЛІННЯ КОРИСТУВАЧАМИ, ДОКУМЕНТАМИ, ЗАВДАННЯМИ І МОЖЛИВОСТІ СПІЛЬНОЇ РОБОТИ

3.1 Реалізація роботи бази даних

Загальна структура бази даних зображено на рисунку 2.5. Весь код для створення бази даних реалізовано мовою SQL, за допомогою запитів до БД.

Зовнішні посилання створено за допомогою команди Reference. Для прикладу код для створення таблиці користувачів (worker), котра має чотири зовнішні ключі які посилаються на таблиці команди (team), роль користувача (role_name), тип посади (job_type_name) та регіон (region_name).

Відповідно таким чином і реалізовано всі решта таблиць. Код для створення таблиці користувачів:

```
DROP TABLE IF EXISTS `worker`;  
  
/*!40101 SET @saved_cs_client      = @@character_set_client */;  
/*!40101 SET character_set_client = utf8 */;  
  
CREATE TABLE `worker` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `birthday` date DEFAULT NULL,  
  `date_hire` date DEFAULT NULL,  
  `login` varchar(255) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `pass` varchar(255) NOT NULL,  
  `phone` varchar(255) DEFAULT NULL,  
  `photo` longblob,  
  `private_mail` varchar(255) DEFAULT NULL,  
  `street` varchar(255) DEFAULT NULL,  
  `surname` varchar(255) NOT NULL,  
  `version` int(11) DEFAULT NULL,  
  `job_type_name` bigint(20) NOT NULL,  
  `region_name` bigint(20) NOT NULL,  
  `role_name` bigint(20) DEFAULT NULL,  
  `team_name` bigint(20) NOT NULL,
```

```

`mobile` varchar(255) DEFAULT NULL,
PRIMARY KEY (`id`),
UNIQUE KEY `login` (`login`),
KEY `FKD162537E30271785` (`team_name`),
KEY `FKD162537EB0DD720C` (`job_type_name`),
KEY `FKD162537E520C2F83` (`role_name`),
KEY `FKD162537EBDECCB25` (`region_name`),
CONSTRAINT `FKD162537E30271785` FOREIGN KEY (`team_name`)
REFERENCES `team` (`id`),
CONSTRAINT `FKD162537E520C2F83` FOREIGN KEY (`role_name`)
REFERENCES `worker_role` (`id`),
CONSTRAINT `FKD162537EB0DD720C` FOREIGN KEY (`job_type_name`)
REFERENCES `worker_job_type` (`id`),
CONSTRAINT `FKD162537EBDECCB25` FOREIGN KEY (`region_name`)
REFERENCES `region` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

```

Посилання на інші таблиці на веб інтерфейсі реалізовано за допомогою випадających списків – це дає можливість забезпечити введення вірних даних і допомагає відобразити вже існуючі в базі даних записи. Для прикладу візьмемо форму для створення нового користувача та вибору регіону, що зображено на рисунку 3.1.

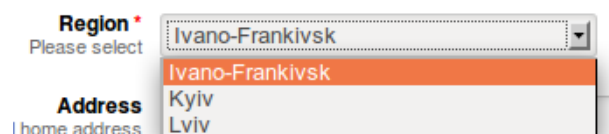


Рисунок 3.1 – Вибір регіону при створенні користувача

3.2 Реалізація веб інтерфейсу

Веб інтерфейс користувача повинний бути зручний та інтуїтивно зрозумілий кожному користувачеві, тому його було реалізовано в легких тонах та зручно розташовано всі навігаційні елементи.

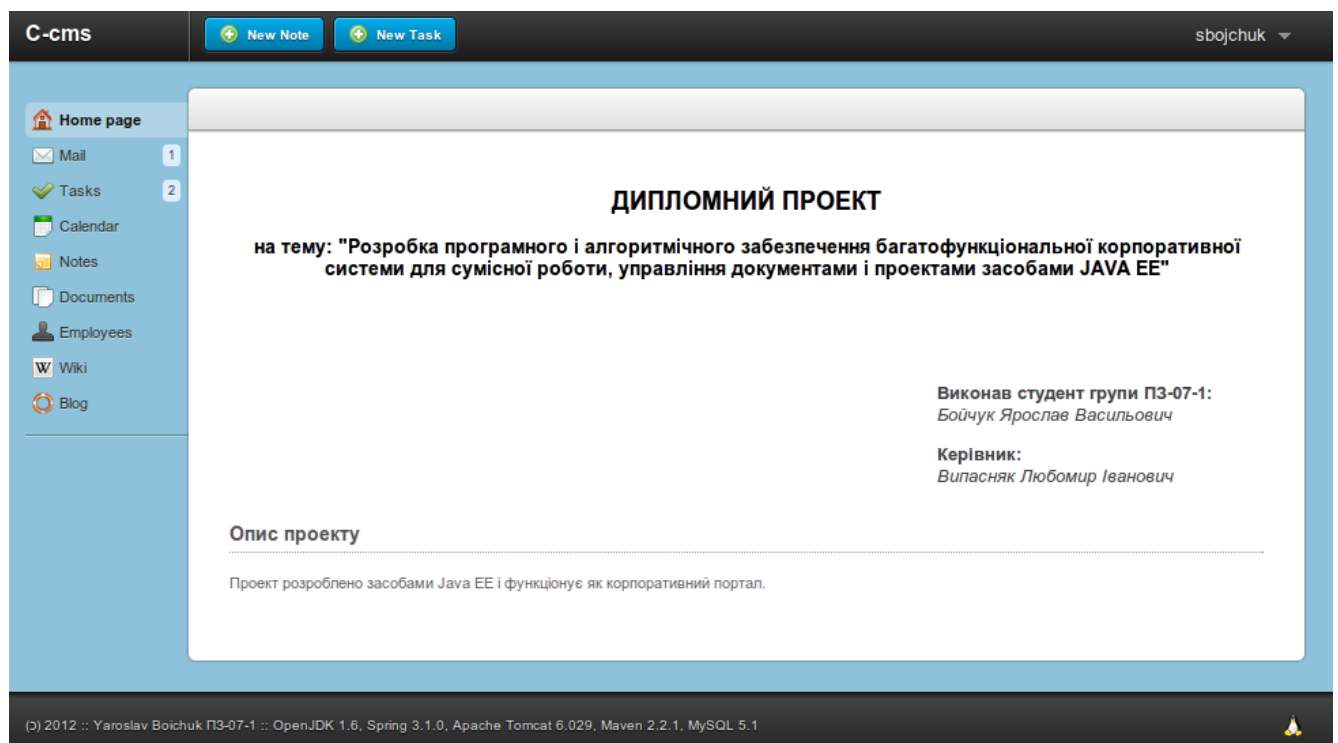


Рисунок 3.2 – Загальний інтерфейс програмного продукту

3.2.1 Навігаційна панель

На навігаційній панелі розташовані елементи швидкого доступу до завдань та задач.

3.2.2 Головне меню

Навігація по веб ресурсу реалізована за допомогою головного меню. В головному меню відображаються всі доступні на сайті навігаційні посилання:

- а) головна сторінка;
- б) корпоративна пошта;
- в) завдання;
- г) календар;
- д) нотатки;
- е) документи;
- ж) список робочих;

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		39