

## ЗМІСТ

<b>ВСТУП</b>	<b>2</b>
<b>1 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЛЯ УПРАВЛІННЯ КОРИСТУВАЧАМИ, ДОКУМЕНТАМИ, ЗАВДАННЯМИ І МОЖЛИВОСТІ СПІЛЬНОЇ РОБОТИ</b>	<b>4</b>
1.1 Реалізація роботи бази даних . . . . .	4
1.2 Реалізація веб інтерфейсу . . . . .	5
1.2.1 Навігаційна панель . . . . .	7
1.2.2 Головне меню . . . . .	8
1.3 Робота із даними . . . . .	10
1.4 Категорії порталу . . . . .	12
1.4.1 Авторизація . . . . .	12
1.4.2 Корпоративна пошта . . . . .	12
1.4.3 Календар . . . . .	13
1.4.4 Завдання і задачі . . . . .	15
<b>ВИСНОВОК</b>	<b>16</b>
<b>ВИКОРИСТАНІ ДЖЕРЕЛА</b>	<b>17</b>

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка програмного алгоритмічного забезпечення багатофункціональної корпоративної системи для сумісної роботи, управління документами і проектами</i> <i>ПОЯСНЮВАЛЬНА ЗАПІСКА</i>	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Розробив</i>	<i>Бойчук Я.В.</i>						<i>1</i>	<i>17</i>
<i>Перевірів</i>	<i>Випасняк Л.І.</i>					<i>ІНФТУНГ ПЗ-07-1</i>		
<i>Т.контр</i>								
<i>Н.контр</i>								
<i>Затв.</i>								

## 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

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

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

## 1.1 Реалізація роботи бази даних

Загальна структура бази даних зображено на рисунку ???. 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 1.1.



The image shows a web form with two fields. The first field is labeled 'Region\*' and has a dropdown menu with 'Ivano-Frankivsk' selected. Below it, the text 'Please select' is visible. The second field is labeled 'Address' and has a text input with 'Kyiv' and 'Lviv' as suggestions. Below it, the text 'I home address' is visible.

Рисунок 1.1 – Вибір регіону при створенні користувача

## 1.2 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.

					ДП.ПЗ 04.00.00.000 ПЗ	Архив
Зм.	Арк.	№ докум.	Підпис	Дата		5

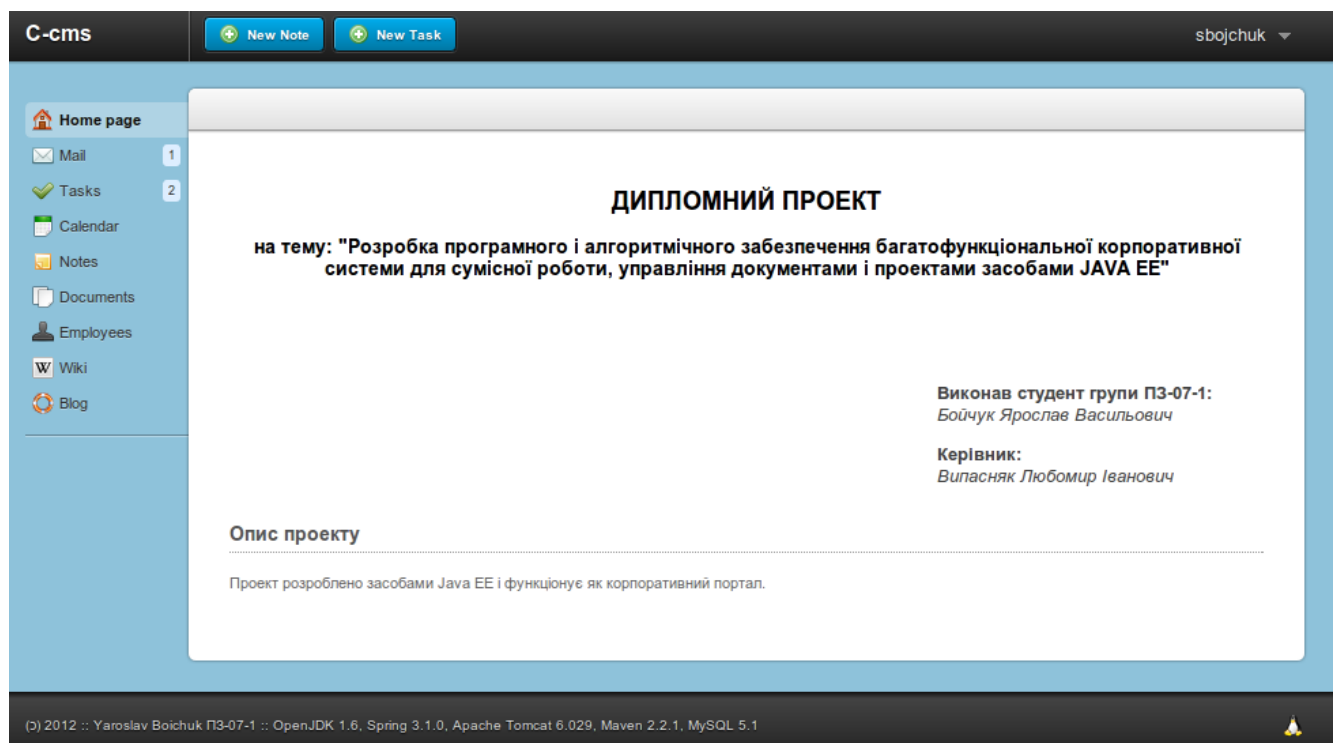


Рисунок 1.2 – Загальний інтерфейс програмного продукту

Весь інтерфейс сайту поділяється на три основні компоненти: головний блок, навігаційна панель, головне меню. Для зручності розробки цих компонентів використано Apache Tiles, що дає змогу поділяти код на логічні одиниці.

```
<body>
<div id="wrapper">
  <header>
    <tiles:insertAttribute name="header" ignore="true" />
  </header>
  <section>
    <div class="container_8 clearfix">
      <tiles:insertAttribute name="menu" ignore="true" />
      <tiles:insertAttribute name="body" />
    </div>
    <div id="push"><!-- --></div>
  </section>
</div>
<footer>
  <tiles:insertAttribute name="footer" ignore="true" />
</footer>
```

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

```

<div class="apple_overlay black" id="overlay">
  <a class="close"></a>
  <iframe class="contentWrap" style="width: 100%; height: 500px
    "></iframe>
</div>
<div style="display: none; position: absolute;" id="calroot">
  <div id="calhead">
    <a id="calprev"></a>
    <div id="caltitle"></div>
    <a id="calnext"></a>
  </div>
  <div id="calbody">
    <div id="caldays">
      <span>Sun</span><span>Mon</span><span>Tue</span><span>Wed</
        span><span>Thu</span><span>Fri</span><span>Sat</span>
    </div>
    <div id="calweeks">
      <!-- -->
    </div>
  </div>
</div>
</body>

```

### 1.2.1 Навігаційна панель

На навігаційній панелі розташовані елементи швидкого доступу до завдань та задач. З легкістю можна додати будь-яке завдання, при чому вибрати заголовок завдання, детальний опис та кінцевий час виконання (рисунки 1.3);

Для зручності навігаційна панель рухається разом із прокруткою сторінки, тобто якщо навіть користувач перейде в низ сторінки, то йому панель буде завжди доступна – це зроблено для простоти і швидкості доступу до створення нової нотатки та завдання.

Справа на навігаційній панелі розташовано меню користувача. Тут знаходяться кнопки переходу на профіль робочого та кнопка виходу із сайту (рисунки 1.4, 1.5);

					ДП.ПЗ 04.00.00.000 ПЗ	Архив
Зм.	Арк.	№ докум.	Підпис	Дата		7

Рисунок 1.3 – Приклад створення завдання із навігаційної панелі

1.4). Після виходу всі дані, які збережені в сесії будуть видалені.

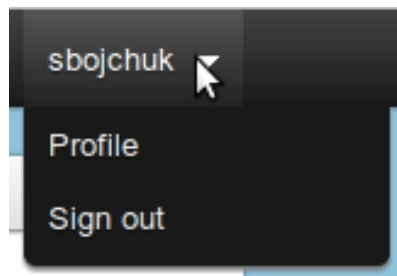


Рисунок 1.4 – Панель користувача

### 1.2.2 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;

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		8



- ж) список робочих;
- и) корпоративна вікі;
- к) блог.

При переході на будь-яке меню, воно зразу підсвічується – це зроблено для зручності користувачеві, щоб було зразу видно де він знаходиться в даний момент часу. Програмно це відбувається за допомогою передачі з контролера в модель атрибута із назвою меню:

```
uiModel.addAttribute("menu", "NOTE");
```

Потім в JSP вигляді головного меню відбувається перевірка на значення поточного меню, і якщо воно сходиться із атрибутом «menu» то додається CSS клас «active» (рисунок 1.5):

```
<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>
```

Відповідно до переходу на певний пункт, відбувається запит контролеру MVC, і вибірка даних із бази даних через контролер з подальшою передачею на вигляд. Для прикладу запит для запису в БД та перевірка на валідність даних:

```
@RequestMapping(method = RequestMethod.POST, produces = "text/
    html")
public String create(@Valid Note note, BindingResult
    bindingResult, Model uiModel, HttpServletRequest
    httpRequest) {
    if (bindingResult.hasErrors()) {
        populateEditForm(uiModel, note);
        uiModel.addAttribute("menu", "NOTE");
        return "redirect:/notes";
    }
    uiModel.asMap().clear();
```

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

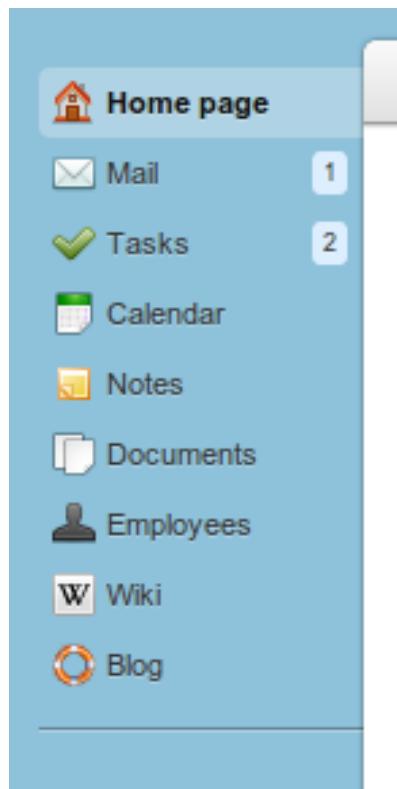


Рисунок 1.5 – Навігаційне меню порталу

```

note.setAuthor(Worker.getPrincipal());
note.setDatetime(new Date());
note.persist();
uiModel.addAttribute("menu", "NOTE");
return "redirect:/notes";
}

```

### 1.3 Робота із даними

Для мапінгу даних із форми до бази даних використовується JPA із Hibernate фреймворком поверх нього. Для кожної форми створюється певний домен (по своїй суті persistence bean), котрий за допомогою анотацій із пакету Javaх дає змогу переносити об'єкти Java в базу даних (за допомогою використання мови запитів Hibernate Query Language). Для прикладу bean для запису нотаток в базу даних:

```

@Configurable
@Entity
public class Note {

```

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

```

@NotNull
private String title;

@NotNull
@Size(max = 1000000)
private String text;

@Temporal(TemporalType.TIMESTAMP)
@DateTimeFormat(style = "M-")
private Date datetime;

@ManyToOne
private Worker author;

public static TypedQuery<Note> findNotesByAuthorEquals(Worker
    author) {
    if (author == null)
        throw new IllegalArgumentException("The author argument
            is required");
    EntityManager em = Note.entityManager();
    TypedQuery<Note> q = em.createQuery("SELECT o FROM Note AS o
        WHERE o.author = :author ORDER by o.id DESC", Note.class);
    q.setParameter("author", author);
    return q;
}

public String getTitle() {
    return this.title;
}

public void setTitle(String title) {
    this.title = title;
}

```

					ДП.ПЗ 04.00.00.000 ПЗ	Архив
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

В вище наведеному кодї кожне поле має свій метод на getter та setter, що дає змогу в вибірці даних, та статичний метод «findNotesByAuthorEquals» для пошуку повідомлень даного автора, що дає змогу в будь-якому місці здійснити операції щодо пошуку цих повідомлень.

## 1.4 Категорії порталу

### 1.4.1 Авторизація

Сторінка авторизації пропонує користувачеві ввести свій логін та пароль (рисунки 1.6) та у випадку неправильних даних буде відображена помилка (рисунки 1.7)

Рисунки 1.6 – Форма авторизації користувача

Рисунки 1.7 – Помилка авторизації користувача

Якщо введено вірні дані, то відбувається запис нової сесії в пам'ять та перенаправлення користувача на головну сторінку, або на сторінку із якої прийшов користувач.

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

### 1.4.2 Корпоративна пошта

Всі отримані листи користувач може переглянути в пункті пошта. Напроти меню «пошта» відображається кількість не прочитаних повідомлень (рисунок 1.8).

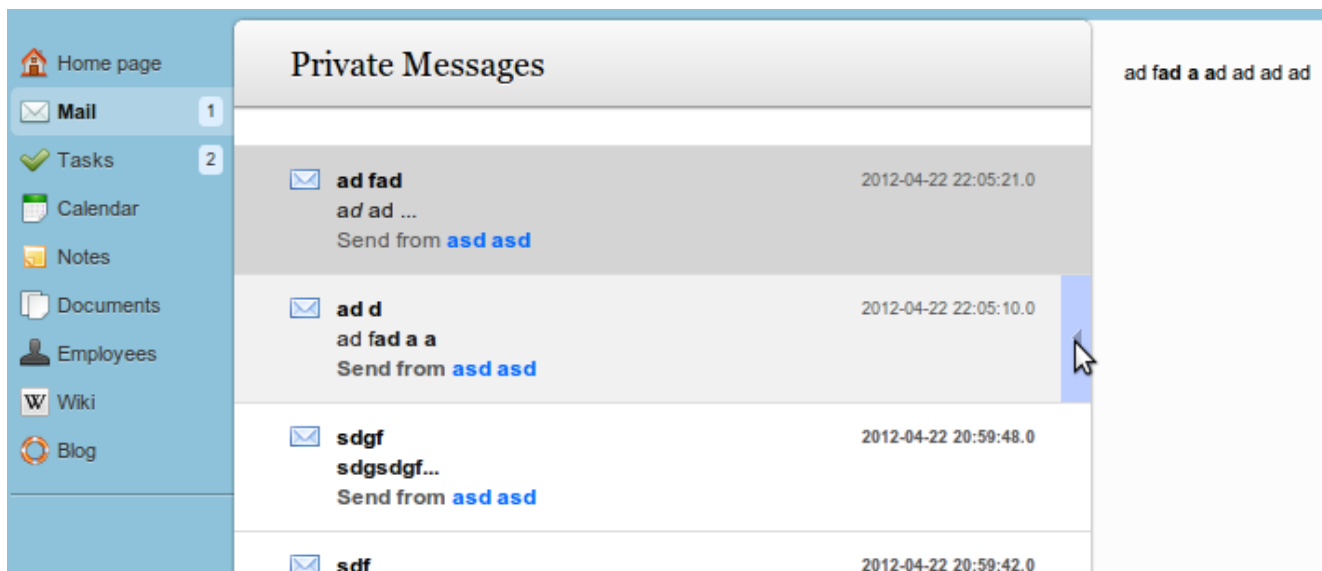


Рисунок 1.8 – Сторінка зі списком повідомлень

Над кожним повідомленням показано тему повідомлення. Головний текст повідомлення обрізаний, проте коли клікнути на повідомлення, збоку висунеться панель із повним описом повідомлення. Також під повідомленням показано час відправлення повідомлення та відправник повідомлення. При кліку на відправника, відбудеться перехід на його персональну сторінку. Кожне не прочитане повідомлення виділяється сірим кольором, для того щоб легше було його знайти, і при детальному перегляді його, колір забереться, і кількість повідомлень, що показуються біля меню – буде зменшено.






Відправлення повідомлення можливе із персональної сторінки кожного користувача. Після переходу на персональну сторінку, слід надрукувати тему повідомлення та саме повідомлення (рисунок 1.9).

Зразу також доступний wysiwyg редактор і live перегляд повідомлення яке друкується. Доставка повідомлення відбувається миттєво, адже використовується локальний сервер бази даних.

					ДП.ПЗ 04.00.00.000 ПЗ	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		13

## Send private message to asd asd

Title \* Нове повідомлення

**B I**   |   

Текст `<em>нового</em>` `<strong>повідомлення</strong>`

Send message

Рисунок 1.9 – Форма для відправлення повідомлень

### 1.4.3 Календар

Персональний календар дає змогу показати всі занесені до нього нотатки та записи. Перегляд даних можливий у трьох проміжних режимах: на місяць, на тиждень (рисунок 1.10) та на день.

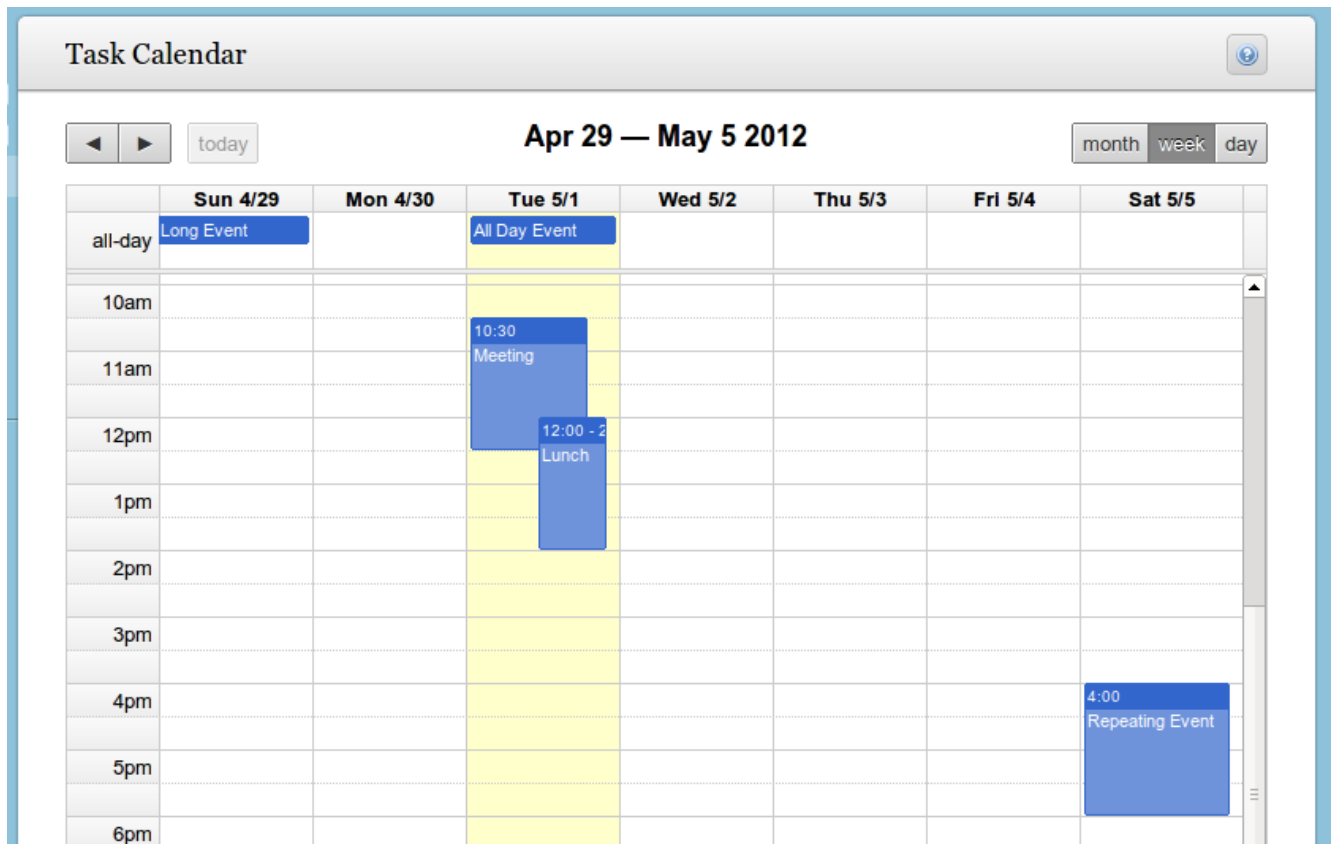


Рисунок 1.10 – Завдання на тиждень

Завдання можуть бути додані як на певний проміжний період, так і на цілий день. Для маніпуляції записів в календарі використана технологія drag & drop від jQuery.

#### 1.4.4 Завдання і задачі

Категорія задач створена для збереження своїх задач (1.3) і можливістю їх перегляду в майбутньому. Це дає змогу всі свої важливі завдання тримати в одному місці (1.11).

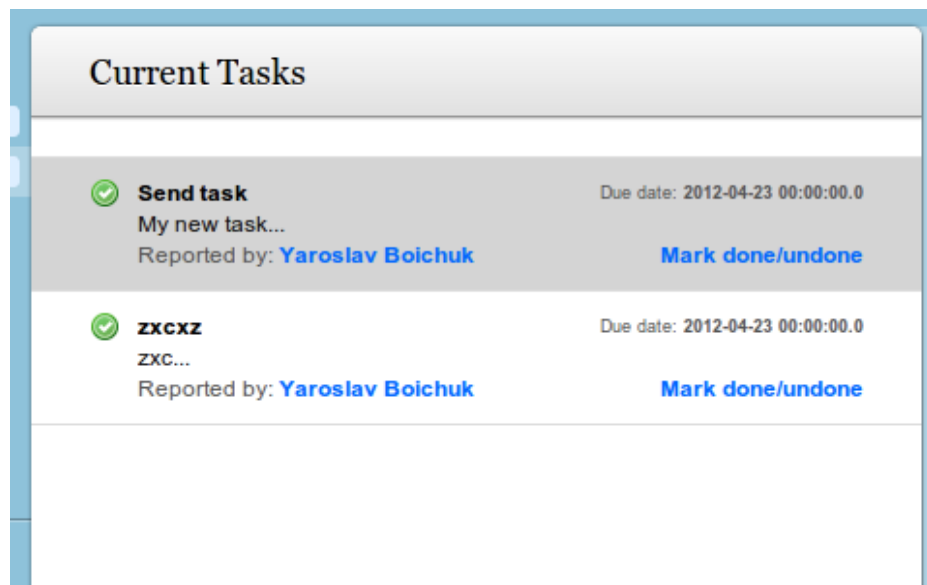


Рисунок 1.11 – Поточні завдання

Кожне завдання яке не виконано ще, позначається аналогічно до непрочитаного повідомлення – сірим кольором, це дає змогу зразу побачити всі поточні завдання. Біля кожного завдання вказано хто створив дане завдання та кінцевий час його виконання. Також в головному меню навпроти пункту «завдання» вказується кількість невиконаних на даний момент завдань. Також кожне завдання може бути позначене як виконане або ж невиконане.

## ВИСНОВОК

Завдяки сучасним технологіям і корпоративним стандартам, розвиток розробки комерційних продуктів виріс дуже стрімко. Зокрема сюди і відноситься відносно молодий напрямок — це розробка корпоративних порталів. Було встановлено стандарти щодо розробки додатків і аплікацій – це допомогло добитися легкої інтеграції і взаємодії. Також проведено аналіз сучасного стану і потреб ринку в даній сфері, наведено всі вимоги до програмного продукту. Проведено аналіз щодо економічної вигоди та прораховано всі важливі аспекти щодо охорони праці.

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



## ВИКОРИСТАНІ ДЖЕРЕЛА

1. <http://www.jcp.org/en/jsr/detail?id=286> - стандарт портлетів Java Portlet 2.0
2. <http://www.jcp.org/en/jsr/detail?id=168> - стандарт портлетів Java Portlet 1.0
3. <http://google.com> - пошук доступної в інтернеті інформації
4. <http://tomcat.apache.org/tomcat-5.5-doc/> – специфікація серлетів
5. <http://pz.nung.edu.ua/> - сайт кафедри ПЗАС
6. <http://www.intranetno.ru/> - бізнес рішення на базі SaaS, PaaS
7. [http://en.wikipedia.org/wiki/Entity-attribute-value\\_model](http://en.wikipedia.org/wiki/Entity-attribute-value_model) - EAV модель

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						17
Зм.	Арк.	№ докум.	Підпис	Дата		