ЗМІСТ

**ПЕРЕЛІК ОСНОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ** 2

ВСТУП 3

*1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ СИСТЕМ КЕРУВАННЯ МЕТАКОНТЕНТОМ ДАЙДЖЕСТОВОГО РЕСУРСУ ТУРОПЕРАТОРА* 4

*1.1 Актуальність проблеми* 4

*1.2.Засоби реалізації системи керування метаконтентом дайджестового ресурсу туроператора* 4

*1.2.1 HTML* 4

*1.2.2CSS* 5

*1.2.3 Javascript* 7

*1.2.5 Spring framework* 10

*1.2.6 Hibernate/Mysql* 20

*1.2.7 Apache Maven* 22

*1.2.10 Google maps* 26

*1.2.11 Panoramio* 27

*1.2 12 Flickr* 28

*1.3 Огляд існуючих засобів керування метаконтентом дайджествого ресурсу туроператора* 29

*1.3.1 Огляд системи travel.ru* 29

*1.3.2.Огляд системи Art-travel* 31

2.ОПИС СЕРВЕРНОЇ ФУНКЦІОНАЛЬНОСТІ СИСТЕМИ КЕРУВАННЯ МЕТАКОНТЕНТОМ ДАЙДЖЕСТОВОГО РЕСУРСУ ТУРОПЕРАТОРА 34

*2.1 Опис функціональних можливостей та структури бази даних системи керування ментаконтентом дайджестового ресурсу туроператора* 34

3.РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КЕРУВАННЯ МЕТАКОНТЕНТОМ ДАЙДЖЕСТОВОГО РЕСУРСУ ТУРОПЕРАТОРА 44

*3.1 Опис програмного продукту* 44

4.ОХОРОНА ПРАЦІ 57

*4.1.Законодавче та нормативно-правове забезпечення охорони праці* 57

*4.2.Організація роботи з охорони праці на підприємстві* 59

*4.3.Основні причини профзахворювань і заходи щодо їх попередження* 66

ВИСНОВКИ 68

Список використаної літератури 69

**ПЕРЕЛІК ОСНОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ**

СRUD- копіювати , змінити,прочитати, видалити . Використовується для позначення операцій над певними об'єктами

Логін - алфавітно-цифровий набір символів, що ідентифікує користувача комп'ютера.

MySQL - система керування реляційними базами даних.

Аватар - фотографія або малюнок, яке користувач використовує разом з псевдонімом.

CSS- каскадні таблиці стилів,які використовуються для задання зовнішнього вигляду веб-сторінки.

AOP- аспектно-орієнтоване програмування  
JDBC -  з'єднання з базами даних на Java  
DAO- шар для зв'язку з БД в аплікації

ORM- об’єктно-реляційне відображення

MVC- модель-вид-контролер . Використовується для розділення бізнес логіки від виду аплікації

JPA -  стандартизований інтерфейс для Java ORM фреймворків .

SQL — мова структурованих запитів

# ВСТУП

На даний момент реклама туристичних послуг стала класичним напрямком реклами в Інтернеті. Туризм , реклама та інтернет нерозривно пов'язані один з одним. Ринок туризму в інтернеті безперервно еволюціонує , освоює нові технічні можливості , створює і розширює рекламні можливості.

Споживач туристичного ринку відчуває величезну потребу в інформації ( докладні описи , фотографії, відгуки ) , яку не здатні задовольнити ні друковані видання , ні тим більше такі засоби масової інформації як телебачення і радіо. Реклама турфірми і просування туристичних послуг стали особливо необхідні в останнє десятиліття , у зв'язку з швидким зростанням туристичного ринку і появою безлічі туроператорів і турагентств. Усе складніше стало рекламувати туристичні послуги та туристичні компанії , все більше потрібно як фінансових ресурсів , так і людських зусиль для залучення клієнтів і продажу туристичного продукту .

Для невеликих туристичних компаній реклама в мережі інтернет цілком може бути єдиним способом донесення інформації до споживачів . Виникає необхідність в рекламі туроператора за допомогою реклами , створення і просування власного сайту підприємства, як одних з найбільш ефективних засобів просування послуг.

# *1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ СИСТЕМ КЕРУВАННЯ МЕТАКОНТЕНТОМ ДАЙДЖЕСТОВОГО РЕСУРСУ ТУРОПЕРАТОРА*

# *1.1 Актуальність проблеми*

Загальновідомий факт , що ефективність прямої реклами туристичних послуг в засобах масової інформації знижується. Пошук альтернативних форм реклами приводить фірми до думки про необхідність власної присутності у всесвітній комп'ютерній мережі Інтернет. Однак успішний вихід в Інтернет досягається далеко не завжди , і в багатьох випадках витрати можуть не виправдатися. Є чимало фірм , які витратили великі кошти на створення чудового по дизайну сайту, але не отримали від цих коштів ніякої віддачі . В результаті - повне розчарування в Інтернеті як технології .

Першим і одним з найважливіших кроків є створення сайту. Зараз вже важко уявити ефективну роботу туристичної фірми без власного сайту. Для туроператора це виключно зручний канал зв'язку з агентствами , для турагента - швидкий вихід на клієнтів.

Актуальність теми також обумовлена ​​тим , що в умовах жорсткої конкуренції , невеликі турфірми можуть бути витіснені з ринку через відсутність правильної політики просування , тому існує необхідність у створенні сайту , щоб залучати нову аудиторію і вести облік відвідування та рейтингу компанії.

# *1.2.Засоби реалізації системи керування метаконтентом дайджестового ресурсу туроператора*

# *1.2.1 HTML*

HTML (від англ. Hypertext Markup Language - мова розмітки гіпертексту) - це стандартна мова розмітки документів у Всесвітній павутині. Всі веб-сторінки створюються за допомогою мови HTML (або XHTML)[2]. Мова HTML інтерпретується браузером і відображається у вигляді документа, зручному для людини. HTML є додатком SGML (стандартної узагальненої мови розмітки) і відповідає міжнародному стандарту ISO 8879.

HTML впроваджує засоби для:

* створення структурованого документу шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
* отримання інформації із Всесвітньої мережі через гіперпосилання;
* створення інтерактивних форм;
* включення зображень, звуку, відео, та інших об'єктів до тексту.

HTML-документ є текстовим файлом розмічений за допомогою спеціальних (текстових) команд. Текстовий формат представлення веб-документів був вибраний виходячи з основних вимог до веб-документу: простота, можливість безпосередньої інтерпретації в будь-якій операційній системі, мінімальний розмір файлу, зручність редагування і інтерпретації.   
Мова розмітки гіпертекстових документів HTML дозволяє визначити різні типи елементів ( у оригіналі element ), що забезпечують функціональність документа: текстові фрагменти із заданими параметрами форматування, списки, таблиці, зображення, гіперпосилання і т.д. Елементи HTML оголошуються за допомогою команд розмітки, званих тегами (від англійського tag - ярлик). Всі HTML-теги, що зустрічаються в тексті документа інтерпретуються браузером при відображенні документа.

# *1.2.2CSS*

Каскадні таблиці стилів (англ. Cascading Style Sheets) — спеціальна мова, що використовується для запису оформлення сторінок, написаних мовами розмітки даних.

Таблиці стилів дають змогу спростити процес створення сторінок і поліпшити їхній зовнішній вигляд[2]. Концепція стилів подібна до ідеї стилів, яка реалізована в сучасних текстових редакторах — текст спочатку вводять, а потім форматують, користуючись стилями. Застосування стилів дає змогу вводити на сторінку потрібні тексти та інші елементи, не задумуючись над їхнім зовнішнім виглядом і розташуванням.

Таблиці стилів програміст зазвичай створює окремо від html-файлу. Під час створення html-файлу він концентрує увагу на змісті сторінки, а не на її зовнішньому вигляді, а під час створення таблиці стилів — навпаки. Отже, стилі дають змогу розмежувати етапи створення html-файлу й удосконалення зовнішнього вигляду сторінки.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

Переваги

* Інформація про стиль для усього сайту або його частин може міститися в одному .css-файлі, що дозволяє швидко робити зміни в дизайні та презентації сторінок;
* Різна інформація про стилі для різних типів користувачів: наприклад великий розмір шрифту для користувачів з послабленим зором, стилі для виводу сторінки на принтер, стиль для мобільних пристроїв;
* Сторінки зменшуються в об'ємі та стають більш структурованими, оскільки інформація про стилі відділена від тексту та має певні правила застосування і сторінка побудована з урахуванням їх;
* Прискорення завантаження сторінок і зменшення обсягів інформації, що передається, навантаження на сервер та канал передачі. Досягається за рахунок того, що сучасні браузери здатні кешувати (запам'ятовувати) інформацію про стилі і використовувати для всіх сторінок, а не завантажувати для кожної.

# *1.2.3 Javascript*

JavaScript — мова програмування для створення інтерактивних Web-сторінок

Найчастіше мова JScript використовується для обробки подій на веб-сторінці таких як натиснення на кнопку відправки форми, вибір нового елементу зі списку, позиціонування курсору мишки над якимось елементом, завершення завантаження сторінки тощо. Призначення такого обробника подій - попередити користувача про можливу помилку в його діях чи створити більш зручну обстановку для роботи зі сторінкою[3]. Проте іноді застосування мови скриптів може бути й більш прозаїчним : в момент завантаження виводити на сторінку HTML-код , що сформує той чи інший елемент сторінки після її завантаження чи провести певні обрахунки.

Не зважаючи на схожість назв, JavaScript та мова програмування Java мають дуже мало спільного. Права на назву JavaScript належать компанії Sun Microsystems.

Розроблена в компанії Netscape. На теперішній час підтримується більшістю браузерів. Текст програми включається безпосередньо в HTML-документ і інтерпретується самим браузером[3]. Застосовується в основному для часткової автоматизації обробки даних, які застосовуються сторінкою.

JavaScript має низку властивостей об’єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об’єктів в ній відрізняється від традиційних мов ООП.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою Сі має такі корінні відмінності:

* об’єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів
* функції як об’єкти першого класу
* обробка винятків
* автоматичне приведення типів
* автоматичне прибирання сміття
* анонімні функції

*1.2.4.Javascript бібліотека Jquery*

jQuery - бібліотека JavaScript, що фокусується на взаємодії JavaScript та HTML. Бібліотека jQuery допомагає легко отримувати доступ до будь-якого елементу DOM, звертатися до атрибутів і вмісту елементів DOM, маніпулювати ними. Також бібліотека jQuery надає зручний API для роботи з AJAX. Автором цієї бібліотеки вважається Джон Резіг[4].

Jquery має такі можливості:

* Перехід по дереву DOM, включаючи підтримку XPath як плагіна;
* події;
* Візуальні ефекти;
* AJAX-доповнення;
* JavaScript-плагіни.

Розробники JavaScript- бібліотеки jQuery прийняли рішення відмовитися від практики подвійного ліцензування і замість поставки коду одночасно під ліцензіями GPL та MIT , продовжити поширення бібліотеки тільки під ліцензією MIT , що не вимагає обов'язкового відкриття коду внесених змін. Зміни ліцензійної політики торкнуться всіх елементів ,що розвиваються в рамках проекту, в тому числі jQuery Core , jQuery UI , jQuery Mobile , Sizzle і QUnit . Причиною відмови від подвійного ліцензування є бажання уникнути плутанини і спростити управління кодовою базою. jQuery Foundation пояснила , що ліцензія MIT є сумісною з GPL і у користувачів є можливість після внесення змін до поставляється під ліцензією MIT проект , переліцензіровать його під ліцензією GPL. Ще однією зміною в політиці проекту є введення в практику підписання членами команди розробників JQuery угоди про передачу організації jQuery Foundation майнових прав на код ( Copyright Assignment Agreement ) , а сторонніми учасниками угоди про ліцензування ( Contributor License Agreement ) , яке вказує на те , що роботодавець дав свою згоду на участь працівника у проекті. Зазначені угоди дозволять гарантувати , що всі права на проект будуть в руках jQuery Foundation і захиститися від претензій компаній роботодавців учасників розробки jQuery (наприклад , якщо працівник займався поліпшенням jQuery в робочий час , компанія може заявити свої права на створений працівником код ) . Крім того , організація jQuery Foundation зможе самостійно приймати пов'язані з інтелектуальною власністю рішення, наприклад , про зміну ліцензії на код , без попереднього схвалення змін у всіх розробників , чий код входить до складу продукту .

# *1.2.5 Spring framework*

Spring - універсальний фреймворк з відкритим вихідним кодом для Java-платформи. Також існує інша версія для платформи. NET Framework, названий Spring.NET.

Перша версія була написана Родом Джонсоном, який вперше опублікував її разом з виданням своєї книги «Expert One-on-One Java EE Design and Development» .

Фреймворк був вперше випущений під ліцензією Apache 2.0 license в червні 2003 року. Перший стабільний реліз 1.0 був випущений в березні 2004. Spring 2.0 був випущений в жовтні 2006, Spring 2.5 - в листопаді 2007, Spring 3.0 в грудні 2009, і Spring 3.1 в грудні 2011.

Незважаючи на те, що Spring Framework не забезпечував якусь конкретну модель програмування, він став широко поширеним в Java-товаристві головним чином як альтернатива і заміна моделі Enterprise JavaBeans. Spring Framework надає велику свободу Java-розробникам в проектуванні, крім того, він надає добре документовані і легкі у використанні засоби вирішення проблем, що виникають при створенні додатків корпоративного масштабу.

Тим часом, особливості ядра Spring Framework застосовні в будь-якому Java-додатку, і існує безліч розширень і удосконалень для побудови веб-додатків на Java Enterprise платформі. З цих причин Spring придбав велику популярність і визнається розробниками як стратегічно важливий фреймворк.

Розглянемо детальніше принципи фреймворка Spring.

**Впровадження залежностей**

Впровадження залежностей є одним з основоположних принципів сучасних JavaEE контейнерів. Детально цей підхід описаний в одній зі статей визнаного гуру Мартіна Фаулера ( Martin Fowler ) .

В даний час існує і продовжує виникати велика кількість програмних продуктів , в основі яких лежить можливість конфігурування залежностей між компонентами прикладної програмної системи не на етапі розробки , а під час її запуску або взагалі вже в процесі функціонування [5]. У результаті такого підходу забезпечення взаємозв'язку різних складових частин системи не залежить від конкретної технології і на етапі тестування не складає труднощів замінити заглушками деякі об'єкти , залежні від нижнього шару аплікації.

Спочатку реалізацією цього підходу займалися творці відкритого програмного забезпечення. Цей принцип наклав також свій відбиток і на створення нових технологій , наприклад , експерти Java Community Process не могли не використовувати цей засіб для зменшення залежностей між компонентами системи при створенні нового стандарту JSR 220 . При цьому у більшості розробників програмного забезпечення, особливо зі світу Java , згадка про впровадження залежностей все таки в першу чергу асоціюється зі Spring Framework , оскільки використання Spring стало фактично стандартом при розробці J2EE додатків .

Якщо в традиційних системах компонент отримує пряме посилання на місцезнаходження необхідних для роботи об'єктів або сервісів , або звертається до сервісу і запрошує посилання на реалізацію певного типу сервісу , то в разі DI контейнера оточення на основі конфігураційних даних , які можуть бути надані системи у вигляді XML-файла або анотацій , саме знає про необхідні взаємозв'язки між компонентами , і надає потрібні об'єкти під час ініціалізації або виконання. Використання впровадження залежності вносить додаткову гнучкість в систему , оскільки полегшується створення альтернативних реалізацій сервісу , і дозволяє вказувати в конфігурації , яка саме реалізація повинна бути використана .

**Аспектно-орієнтоване програмування ( AOP )**

Аспектно-орієнтоване програмування, АОП (англ. aspect-oriented programming, AOP)  — парадигма програмування, яка дозволяє виокремити перехресну (наскрізну) функціональність[4].

Сучасні програмні системи часто вирішують величезну кількість складних завдань, що потребують хороших інженерних навичок від їх розробників та надійності інструментальних засобів розробки. При зростанні складності таких систем зростає і програмний код, розробнику стає все важче охопити всі деталі реалізації системи. При підтримці великих програмних засобів зростає час знаходження та виправлення помилки, ускладнюється додавання нових характеристик, оскільки стає все важче визначити наскільки зміни вплинуть на систему, чи не внесуть додаткових помилок та дефектів. Для вирішення таких завдань застосовують різноманітні інженерні засоби, як от багатофункціональні середовища розробки, шаблони

проектування готові програмні каркаси тощо.

Часто згадуваним недоліком об’єктно-орієнтованого підходу є неможливість локалізації наскрізної функціональності в одному класі. Як приклад такої функціональності часто називають необхідність ведення журналів подій, керування винятковими ситуаціями, перевірку прав доступу. Код, що відповідає за дану функціональність, часто розкиданий по різних класах. Це, з одного боку, не дозволяє сконцентрувати увагу на основній бізнес-логіці класу і ускладнює читання коду. З іншого боку, ускладнюється внесення змін у методи роботи наскрізної функціональності, що не завжди можна виправити правильним використанням інтерфейсів чи шаблонів проектування. Наразі аспектно-орієнтований підхід часто використовують для реалізації вищенаведених прикладів, проте, як зауважують деякі автори, на цьому сфера застосування аспектно-орієнтованого підходу не обмежуються, оскільки він може бути використаний для проектування будь-яких систем, що містять наскрізну функціональність ,яка може бути розбита на етапи.  
 В результаті наявності зайвої перехресної функціональності модуль ,що розробляється містить заплутаний код, що задовольняє різні програмні вимоги. Негативні властивості такого коду:

* Розкиданий код. Оскільки наскрізна функціональність зачіпає багато модулів системи, то й виклики цієї функціональності будуть розкидані по всій системі. Наприклад, якщо програма містить засоби моніторингу продуктивності роботи з базою даних, то й виклики цієї функціональності будуть розміщені всюди, де потрібно працювати з базою даних. Наявність розкиданого коду має негативний вплив на проектування та реалізацію системи;
* Важкість відслідковування призначення модуля, оскільки він містить одночасно функціональність для задоволення різних вимог;
* Складність або неможливість повторного використання модуля у програмах з іншою наскрізною функціональністю;
* Велика ймовірність помилок. Наявність коду для реалізації функціональності різного роду в одному модулі може призвести до того, що жодне із завдань не отримає достатньої уваги розробника;
* Важкість супроводу. Якщо з'являється необхідність у зміні наскрізною функціональності, така зміна зачіпає багато модулів системи, що в кінцевому рахунку може призвести до проблем сумісності.

Для вирішення завдання локалізації наскрізної функціональності була розроблена методологія аспектно-орієнтованого програмування (АОП). Основні ідеї АОП були сформульовані ідеологом методології Г. Кінжалесом. Він також розробив найпопулярнішу надбудову мови програмування Java для роботи з аспектами – AspectJ.

До основних понять аспектно-орієнованого програмування належать[5]:

Аспект (англ. aspect) — модуль або клас, який реалізує наскрізну функціональність. Аспект змінює поведінку іншого коду, застосовуючи поради в точках з'єднання, визначених деяким зрізом;

Порада (англ. advice) — додаткова логіка, код, який повинен бути викликаний з точки з'єднання. Порада може бути виконана до, після або замість точки з'єднання;

Мета (англ. target) — об'єкт, до якого будуть застосовуватися поради;

Точка з'єднання (англ. join point) — точка в виконуваній програмі (виклик методу, створення об'єкта, звернення до змінної), де слід застосувати пораду;

Зріз (англ. pointcut) — набір точок з'єднання. Зріз визначає, чи підходить дана точка з'єднання до заданої поради;

Впровадження (англ. introduction) — зміна структури класу та/або зміна ієрархії успадкування для додавання функціональності аспекту в чужорідний код;

Переплетення (англ. weaving) — зв'язування об'єктів з відповідними аспектами (можливе на етапі компіляції, завантаження або виконання програми).

Аспектно-орієнтований ( AOP ) Spring - поставляється з багатою підтримкою аспектно-орієнтованого програмування ( AOP ) , що робить можливим розбити процес виконання програми , відокремлюючи бізнес-логіку програми від системних сервісів (таких як аудит та управління транзакціями ) . Об'єкти програми роблять те , що повинні робити - виконують прикладну логіку і нічого більше. Вони не відповідають за інші системні питання (або навіть не інформовані про них) , такі як логування ( logging ) або підтримка транзакцій .

Всі мови АОП надають кошти для виділення наскрізний функціональності в окрему сутність. Так як AspectJ є родоначальником цього напрямку, використовувані в цьому розширенні концепції поширилися на більшість мов АОП.

Способи застосування

Використання аспектно-орієнтованого підходу не вимагає повністю відмовитись від об’єктно-орієнтованої реалізації, оскільки його можна впроваджувати лише частково. Більше того, такий підхід ефективно доповнює об'єктно-орієнтований код. Об'єктно-орієнтований програмний каркас містить компоненти, що становлять ядро функціонування та компоненти, що містять додаткову функціональність. При використанні фреймворку стандартна функціональність розширюється за допомогою наслідування. Застосування аспектно-орієнтованого підходу дозволяє, з одного боку, розділити програму на окремі модулі , з іншого боку — легко додавати функціональність використовуючи аспекти AOP підходу до проекта.  
Ефективно можна застосовувати аспектно-орієнтоване програмування для оптимізації шаблонів проектування. Першою значною перевагою є здатність локалізувати код шаблону проектування в одному аспекті або парі тісно пов'язаних аспектів . Запропоновані методи застосування аспектно-орієнтованого підходу для розробки багатоагентних систем , де агент – автономна програмна одиниця, що при виконанні завдання реагує на навколишнє середовище та спілкується з іншими програмами-агентами (наприклад, програми купівлі товарів в Інтернеті). В даному випадку аспекти можна застосувати для проектування кількох платформ комунікації агентів, додаткових можливостей навчання, протоколів взаємодії.

**Модулі Spring**

Spring фреймворк складається з декількох структурованих модулів .У сукупності дані модулі надають розробнику все необхідне для проектування додатків високого рівня[6]. Але розробники не зобов'язані створювати аплікацію тільки на фреймворку Spring . Вони вільні у виборі модулів , які необхідні у додатку , і пошук альтернатив у тому випадку , якщо Spring не відповідає всім вимогам. Фактично Spring пропонує точки інтеграції з деякими іншими фреймворками та бібліотеками , і розробниками не доведеться писати їх.

Як можна бачити , всі модулі Spring(рис.1) побудовані на базі ядра контейнера. Контейнер , будучи істотною частиною Spring , визначає , як компоненти створюються , конфігуруються і управляються . Розробнику можна неявно використовувати ці класи , коли відбувається конфігурація програми. Також можна використовувати інші модулі ,включаючи модулі для інтеграції з іншими технологіями. Ці модулі забезпечать зв’язок з фреймворками , за допомогою можна будувати додаток з допомогою таких сервісів як AOP і Persistence.

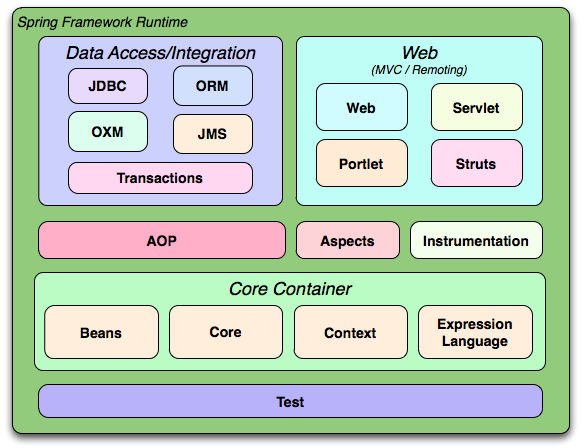


Рис.1.1 Компоненти Spring framework

**Контейнер**

Контейнер Spring є контейнером в тому сенсі , що він містить у собі об’єкти і управляє життєвим циклом і конфігурацією програмних об'єктів. У Spring можна визначити порядок створення програмних об'єктів , їх конфігурацію , і те , як вони взаємодіють один з одним. Spring Framework дозволяє конфігурувати і складати додатки з простих компонентів.

У Spring програмні об'єкти задаються зазвичай в XML файлі але є можливість це зробити за допомогою анотацій . Крім того Spring забезпечує інфраструктурну функціональність (управління транзакцій , взаємодія з фреймворками , зберігання даних і т.д. ), залишаючи розробку прикладної логіки за програмістом.

Ядро контейнера надає основну функціональність фреймворка Spring . Цей модуль складається з BeanFactory , який є основним контейнером Spring і те, на чому грунтується Spring DI .

**Контекстний модуль програми**

Цей модуль Spring будується на ядрі контейнера. BeanFactory в модулі ядра робить Spring контейнером , в той час як контекстний модуль робить його фреймворком . Даний модуль розширює концепцію BeanFactory , додаючи підтримку інтернаціоналізації ( I18N ) повідомлень , подій життєвого циклу програми. Крім того , цей модуль пропонує багато сервісів , такі як email , доступ за допомогою JNDI , EJB інтеграцію , віддалений виклик і запланований запуск. Також включена підтримка інтеграції з фреймворками шаблонів , такими як Velocity і FreeMarker .

**AOP модуль**

Spring забезпечує багату підтримку аспектно-орієнтованого програмування у своєму AOP модулі. Даний модуль працює як базис при розробці аспектів у додатку , побудованому з використанням Spring . Як і DI , AOP забезпечує слабку зв'язаність об'єктів програми. Однак за допомогою AOP програмні компоненти ( такі як транзакції і безпека) відокремлюються від об'єктів , до яких вони застосовуються. Модуль AOP в Spring надає кілька підходів у побудові аспектів , включаючи створення аспектів , заснованих на інтерфейсах AOP Alliance і підтримку AspectJ .

**Модуль абстракції JDBC і DAO**

Робота з JDBC найчастіше зводиться до використання шаблонного коду , який отримує з'єднання , створює SQL вираз , обробляє результат , а потім закриває з'єднання . JDBC і модуль DAO в Spring відокремлюються від шаблонного коду , що дозволить підтримувати код , який стосується доступу до баз даних , прозорим і простим , а також зупинить проблеми , що з'являються в результаті помилки звільнення ресурсів бази даних. До того ж даний модуль використовує AOP модуль , щоб надати сервіси управління транзакціями для об'єктів в додатку створеному за допомогою Spring .

**Модуль інтеграції об'єктно-реляційного відображення ( ORM )**

Для зв'язку з базою даних найчастіше використовують об'єктно- реляційного відображення ( ORM ) поверх JDBC. Підтримка ORM в Spring будується на підтримці DAO , що забезпечує зручний спосіб створення DAO об'єктів для деяких ORM рішень . Spring не намагається реалізувати своє власне ORM рішення , але просто надає важелі управління деякими популярними ORM фреймворками , включаючи Hibernate , Java Persistence API , Java Data Objects , і iBATIS SQL Maps . Spring підтримує можливість управління транзакціями в ORM фреймворки так само, як і JDBC .

**MVC в Spring**

MVC є загальноприйнятим підходом до побудови веб-додатків таким чином , щоб користувальницький інтерфейс був відділений від логіки програми. Java не відчуває нестачі в MVC фреймворках , таких як Apache Struts , JSF , WebWork , і Tapestry , які є найбільш популярним вибором серед MVC . Хоча Spring інтегрується з кількома популярними MVC фреймворками , він також поставляється зі своїм власним дуже потужним MVC модулем.

**Portlet MVC в Spring**

Багато веб -додатків побудовані за принципом сторінок , який полягає в тому , що кожен запит до додатка дає на виході відображення абсолютно нової сторінки. Кожна сторінка звичайно являє специфічний набір інформації або запит нових даних від користувача в певній формі . На відміну від цього , додатки, засновані на Portlet , зосереджують кілька частин функціональності в одну веб-сторінку. Portlet MVC грунтується на Spring MVC і надає набір контролерів , які підтримують Java portlet API .

**Веб модуль в Spring**

Spring MVC і Spring Portlet MVC вимагають особливого підходу при завантаженні контексту програми Spring[7] . Тому веб-модуль Spring надає спеціальні класи для підтримки Spring MVC і Spring Portlet MVC . До того ж веб-модуль пропонує підтримку деяких веб-орієнтованих завдань , таких як завантаження файлів декількома частинами і автоматичне зв'язування параметрів запиту з вашими бізнес-об'єктами . Також він містить підтримку інтеграції з Apache Struts і JavaServer Faces .

**Віддалений виклик**

Не всі програми працюють самостійно. Найчастіше їм необхідно задіяти функціональність інших програм , щоб виконати свою роботу. При доступі по мережі до іншої програми для комунікації використовується деяка форма віддаленого виклику . Підтримка віддаленого виклику в Spring робить доступною функціональність Java об'єктів як віддалених об'єктів. Або , у випадку , коли необхідний віддалений доступ до об'єктів , модуль для віддаленого виклику спрощує компоновку віддалених об'єктів в додатку , ніби вони є звичайними локальними об'єктами Java. Доступно декілька варіантів віддаленого виклику , включаючи Remote Method Invocation , Hessian , Burlap , JAX -RPC і власний HTTP Invoker .

**Java Message Service**

Недолік віддаленого доступу в тому , що він залежить від надійності мережі і вимагає , щоб обидва кінці зв'язку були доступні. Зв'язок , який ґрунтується на повідомленнях більш надійніший і гарантує доставку повідомлень , навіть коли мережа і кінцеві вузли ненадійні. Модуль Spring Java Message Service допомагає відправляти повідомлення в JMS чергу повідомлень. Водночас , цей модуль також допомагає створювати POJO об'єкти, здатний приймати асинхронні повідомлення.

**Spring Data JPA**

Spring Data дозволяє легше створювати Spring - керовані програми які використовують нові способи доступу до даних , наприклад нереляційні бази даних , map-reduce фреймворки , cloud сервіси , а так само вже добре поліпшену підтримку реляційних баз даних. Якщо потрібно швидко в проекті створити шар для зв`язку з базою даних ,який базується на JPA , призначений в основному для CRUD операцій , і нам не потрібно створювати абстрактні DAO класи , інтерфейси ,їх реалізації , то Spring Data - JPA це хороший вибір.

Отже: якщо оголити Spring до рівня базових компонентів , розробник отримує фреймворк , який допомагає йому розробляти слабо зв'язаний програмний код . Навіть якщо це все, що може Spring , переваги слабкого зв'язування ( масштабованість і простота тестування ) роблять Spring популярним фреймворком для створення додатків .А також великий вибір модулів і технологій ,які з ним можуть працювати (Google Web Toolkit,Apache tiles,Jasper Reports ) роблять його потужним інструментом для розробки додатків орієнтованих на веб.

# *1.2.6 Hibernate/Mysql*

Hibernate - бібліотека для мови програмування Java, призначена для вирішення завдань об'єктно-реляційного відображення . Вона являє собою програмне забезпечення з відкритим вихідним кодом (open source)[8], яке розповсюджується на умовах GNU Lesser General Public License. Дана бібліотека надає легкий у використанні фреймворк для відображення об'єктно-орієнтованої моделі даних в традиційні реляційні бази даних.

Метою Hibernate є звільнення розробника від значного обсягу порівняно низькорівневого програмування щодо забезпечення зберігання об'єктів у реляційній базі даних. Розробник може використовувати Hibernate як в процесі проектування системи класів і таблиць «з нуля» , так і для роботи з вже існуючою базою даних.

Hibernate не тільки вирішує завдання зв'язку класів Java з таблицями бази даних ( і типів даних Java з типами даних SQL) , але також надає засоби для автоматичної генерації та оновлення набору таблиць , побудови запитів і обробки отриманих даних і може значно зменшити час розробки , яке зазвичай витрачається на ручне написання SQL - і JDBC - коду. Hibernate автоматизує генерацію SQL - запитів і звільняє розробника від ручної обробки результуючого набору даних і перетворення об'єктів , максимально полегшуючи перенос додатку на будь-які бази даних SQL.

Hibernate забезпечує прозору підтримку збереження даних ( persistence ) для « POJO » . Єдина вимога для класу - наявність конструктора за замовчуванням . Для коректної поведінки в деяких додатках потрібно також визначити методи equals ( ) і hashCode ( ).

Mysql

MySQL– вільна реляційна система керування базами даних, яка працює, як сервер для забезпечення багатокористувацького доступу до великої кількості баз даних.

Проекти на основі безкоштовного ПЗ, які вимагають повнофункціональної системи керування базами даних часто використовують MySQL. До таких проектів відносяться, наприклад, WordPress, phpBB, Drupal та інше програмне забезпечення, побудоване на стеку продуктів LAMP (Linux, Apache, MySQL, PHP/Perl/Python). MySQL також використовується в багатьох гучних великомасштабних Web-продуктах, включаючи Wikipedia, Google (для програми AdWords),  Facebook, YouTube, Flickr, Yahoo!, Digg, LiveJournal, Nokia тощо.

Для некомерційного використання MySQL є безкоштовним. Можливості сервера MySQL:

* простота у встановленні та використанні;
* підтримується необмежена кількість користувачів, що одночасно працюють із БД;
* кількість рядків у таблицях може досягати 50 млн.;
* висока швидкість виконання команд;
* наявність простої і ефективної системи безпеки.
* Крос-платформна підтримка
* Збереження процедур та функцій
* Тригери
* Курсори
* Інформаційна схема (так званий системний словник, що містить метадані).
* Підтримка SSL
* Кешування запитів
* Вкладені запити SELECT
* Підтримка реплікації
* Повноцінна підтримка Юнікоду (UTF-8 і UCS2)
* сегментування таблиць
* тощо.

# *1.2.7 Apache Maven*

Apache Maven - фреймворк для автоматизації збирання проектів, який задає конфігурацію збирання проекту в POM файлі.

Maven, на відміну від головоного конкурента Apache Ant, забезпечує декларативну, а не імперативну збірку проекту. Тобто, в файлах проекту pom.xml міститься його декларативне опис, а не окремі команди. Усі завдання з обробки файлів Maven виконує через плагіни[9].

Maven підтримує принцип «конвенції над конфігурацією» (Convention over Configuration). Оскільки проект дотримується обраної системи конвенцій відпадає необхідність специфікувати їх, що сильно спрощує pom.xml. Однак, майже всі стандарти, на які спирається Maven, можуть бути замінені індивідуальної конфігурацією.

Стандартна структура каталогів - одна з реалізацій цього принципу. Оскільки проект її дотримується - відпадає необхідність вказувати шлях до файлів, що сильно спрощує pom.xml.

Наступна структура показує найважливіші каталоги проекта.

Кореневий каталог проекту: файл pom.xml і всі подальші підкаталоги

src: всі вихідні файли

src / main: вихідні файли власне для продукту

src / main / java: Java - класи

src / main / resources: інші файли, які використовуються при компіляції або виконанні, наприклад Properties-файли

src / test: вихідні файли, необхідні для організації автоматичного тестування

src / test / java: JUnit-тест-завдання для автоматичного тестування

target: створені в процесі роботи Maven файли

target / classes: скомпільовані Maven Java-класи

Життєвий цикл проекту - це список фаз, що визначає порядок дій при побудові проекту. Maven використовує за замовчуванням наступний життєвий цикл:

1. Створення темплейту та обробка ресурсів (archetype): На цій фазі знаходяться і, при необхідності, викачуються з інтернету залежності для роботи проекту.
2. Компіляція (compile)
3. Обробка тестових ресурсів. (Наприклад - скачується з інтернету JUnit-пакет).
4. Компіляція тестів. (Класи тестів не передаються кінцевим користувачам.)
5. Тестування (test)
6. Створення пакета (package). Звичайно мова йде про створення JAR-або WAR-файла.
7. Інсталяція проекту в локальному Maven-репозиторії (install). Тепер він доступний як модуль для інших локальних проектів.
8. Інсталяція в відаленому Maven-репозиторії (deploy). Тепер стабільна версія проекту доступна широкому колу розробників.

Maven має також фази для очищення (cleaning) і для генерації сторінок (site).

Стандартні життєві цикли можуть бути істотно доповнені Maven-плагінами і Maven-архетипами (англ. Archetypes). Maven-плагіни дозволяють вставляти в стандартний цикл нові кроки або розширювати існуючі кроки. Maven-архетипи представляють собою заготовки для різних програмних пакетів (якщо вони відповідають стандартам Maven-структури).

*1.2.8Apache tiles*

Apache tiles дозволяє збирати з фрагментів загальну сторінку. Це дозволяє розбити сторінку на фрагменти і потім знову використати вже готові елементи і лишень змінювати динамічну частину сторінок[10]. Для того щоб створити готову сторінку tiles нам потрібно прописати її в файл xml,який буде відображати структуру нашої веб-сторінки з динамічними і статичними елементами.   
<definition name="index" template="/WEB-INF/view/layouts/classic.jsp">

<put-attribute name="title" value="Welcome,stranger" />

<put-attribute name="header" value="/WEB-INF/view/tiles/header-page.jsp" />

<put-attribute name="body" value="/WEB-INF/view/pages/index.jsp" />

<put-attribute name="footer" value="/WEB-INF/tiles/footer-page.jsp" />

</definition>

<definition name="subcategory" extends="index">

<put-attribute name="title" value="Forum" />

<put-attribute name="body" value="/WEB-INF/view/pages/subcategory.jsp" />

</definition>

Як бачимо в нас є шаблон(template) тобто сторінка на якій ми можемо вставити всі фрагменти на свої місця. Після цього ми можемо наслідувати шаблон з поставленими вже фрагментами(header,footer ) і відповідно створювати нову сторінки tiles лише змінюючи динамічні елементи сторінки(body).

*1.2.9 Twitter Bootstrap*

Bootstrap — це набір інструментів від Twitter (відноситься до класу інструментів: CSS-фреймворк), створений для полегшення розробки web застосунків та сайтів. Він включає CSS та HTML для типографії, форм, кнопок, таблиць, сіток, навігації тощо, а також додаткові розширення для мови JavaScript.

Основні інструменти Bootstrap:

* сітки — наперед задані розміри колонок, які можна відразу ж використовувати, наприклад ширина колонки 90px відноситься до класу .span2, який ми можемо використовувати в CSS описі документа;
* шаблони — Фіксований або ґумовий шаблон документа;
* типографіка — Опис шрифтів, визначення деяких класів для шрифтів таких як код, цитати тощо;
* медіа — Представляє певне управління зображеннями та відео;
* таблиці  — Засоби оформлення таблиць, дозволяє додавати функціональність сортування;
* форми — Класи для оформлення не тільки форм але і деяких подій;
* навігація — Класи оформлення для табів, вкладок, сторінок,
* алерт — Оформлення діалогових вікон, підказок і спливаючих вікон.

Bootstrap використовує найсучасніші напрацювання в області CSS та HTML, тому необхідно бути уважним при підтримці старих браузерів. Він дозволяє легко і швидко зверстати нескладний сайт , маючи мінімальні знання в області верстки html – сторінок[11].

Цей фреймворк здатний істотно полегшити життя тим,хто займається версткою , і вже претендує на звання нового стандарту по верстці інтернет- сторінок.

Bootstrap - це фреймворк від розробників Twitter - Марка Отто і Якоба Торнтона , спочатку планувався як внутрішній продукт, призначений для підтримки одноманітності внутрішніх інструментів. Через якийсь час, у серпні 2011 року фреймворк був випущений для загального користування. Сьогодні Twitter Bootstrap є одним з найпопулярніших html / css фреймворків .

Цей інструмент дозволяє легко і швидко зверстати нескладний сайт , маючи мінімальні знання в області верстки html - сторінок.

Однак , незважаючи на всю привабливість можливостей Bootstrap'а , далеко не всі розробники і ті , кому його можливості могли б знадобитися , вміють користуватися цією технологією. У цій статті ми постараємося розібратися , з чого почати роботу з цим фреймворком .

По суті , Bootstrap містить готові стилі для оформлення основних елементів html , в тому числі кнопок , таблиць , графіків , картинок і відео , форм , елементів навігації , алертів , стилі для заголовків , списків , цитат , визначень , кодів , а також самі базові і часто використовувані динамічні елементи на javascript , такі як нескладний слайдер , що випадають списки , вкладені меню , лайтбокс , пагінацію і т.д. Цей фреймворк дозволяє створювати первісну структуру і стилізувати на базовому рівні більшість елементів . Тим , кому потрібний найпростіший сайт , для верстки , можливо , буде достатньо тільки знання html і того , що робить кожен з класів Bootstrap - часто про це можна здогадатися виходячи вже з їх назви.

# *1.2.10 Google maps*

Google Maps — набір додатків, побудованих на основі безкоштовного картографічного сервісу і технологій, які надає компанія Google.

Існує можливість використовувати сервіс для створення своїх продуктів сторонніми компаніями. На сьогоднішній день це безкоштовна служба, але можливість додати рекламу залишена на майбутнє[12].

Для розробників сайтів зручно буде використати JavaScript для керування функціональністю карт, правда кількість запитів з одного сервера обмежена. Google Static Maps API дозволяє будувати статичні мапи за допомогою спеціальних адресів . Також існують версії  API під різні види мобільних пристроїв.

З сервісом також пов'язаніий додаток Google Earth — окрема програма для Microsoft Windows, а також GNU/Linux, Mac OS. Як і Google Maps, програма Google Earth дозволяє переглядати знімки земної поверхні, змінювати масштаб і будувати маршрути пересування. Її перевагою є тривимірне зображення земної поверхні (з урахуванням рельєфу), можливість спостереження під довільним кутом (а не тільки прямовисно згори), поступове уточнення зображення по мірі завантаження детальніших фотознімків, можливість плавної зміни масштабу.

Google Street View надає користувачам можливість побачити в тривимірній проекції вулиць через інтернет. Така функціональність стала можлива за допомоги кругового фотографування місцевості спеціальним обладнанням в режимі реального часу. В підсумку створюється багато сферичних панорам з прив'язкою до географічних координат та надається можливість перемикатись між ними, використовуючи для цього інтуїтивно зрозумілий інтерфейс, створюючи відчуття пересування в просторі. Є можливість розпізнавати пласкі поверхні на фотографії, такі як фасади будівель, також присутня можливість отримання найвдалішого ракурсу для перегляду вибраного виду.

# *1.2.11 Panoramio*

Panoramio — веб-сайт для розміщення фотографій, що дозволяє зберігати їх географічні координати. Штаб-квартира Panoramio знаходиться в Цюриху, в офісній будівлі швейцарського підрозділу Google. Сервери, що зберігають фотографії користувачів, розташовані в США.

Сайт інтегрований з Картами Google. Це дозволяє як встановити географічне положення об'єктів на фотографіях, завантажених користувачами з усього світу, так і подивитися фотографії місцевості, що переглядається на карті.

З грудня 2006 року зображення з Panoramio доступні на Google Earth при активізації шару Panoramio [13]. Оновлення фотографій з Panoramio на карті Google Earth відбувається кілька разів на тиждень .

Після завантаження фотографії, а також прив'язки до місця, фото розглядається модераторами. Після перевірки (вона може тривати від 1 години до декількох днів) і при дотриманні політики Google Earth, маркується невеликим значком (логотипом Panoramio) і написом внизу фотографії — «фотографія обрана для Google Earth». У Google Earth вона стає видною під час чергового процесу «оновлення» фотографій.

Panoramio надає API для отримання фотографій за допомогою координат місця. В даній дипломній роботі він використаний у поєднанні з Google Maps.

# *1.2 12 Flickr*

Flickr — веб-сайт для розміщення фотографій, їх обговорення та архівування. Flickr є фотосервіс порталу Yahoo!, який широко використовується блоггерами для зберігання фотографій. Популярний завдяки зручній та простій системі завантаження та пошуку фотографій. Дозволяє створювати тематичні групи, соціальні мережі[14].

Кожний, хто зареєструвався на порталах Yahoo! або Google, або є учасником соціальної мережі Facebook автоматично отримує доступ до послуг фотосайту Flickr.

Для зв`язку з сервісом було використано технологію REST і сам сервіс надає REST API за допомогою якого можна виконувати всі функції користувача . В даній програмі було застсосовано java аплікацію ,яка обгортає REST і дозволяє легше і швидше використовувати REST API за допомогою Java.

Власник фотографій має можливість:

* обмежити доступ до своїх фото;
* встановити умови використання фото;
* супроводжувати свої фото коментарями;
* дозволяти коментувати свої фото;
* ставити до кожного фото мітки (теги), які дозволяють знаходити фото в архівах Flickr за темою, місцем або предметом присутніми на фото;
* позначати певні фрагменти фото з коментарем;
* прив'язувати фото до місця зйомки або місцезнаходження об'єкта на географічній карті з великою точністю;
* розмістити кожне своє фото в 10 тематичних групах (для платних користувачів максимальна кількість груп — 60).

# *1.3 Огляд існуючих засобів керування метаконтентом дайджествого ресурсу туроператора*

# *1.3.1 Огляд системи travel.ru*

Сайт містить багато інформації необхідної туристу: опис країн , пошук турів , бронювання готелів , квитків і трансферів.

У розділі " Агентствам " компанії можуть безкоштовно розмістити коротку інформацію про себе. Розміщення турів - на платній основі. Попередні клієнтські заявки з сервера TRAVEL.RU відправляються в агентство по електронній пошті. Тут же їм надається можливість створити власну інтернет- систему бронювання авіаквитків у себе на сайті за допомогою продукту " EasyRes " . Один з найцікавіших розділів сайту - " листи читачів " , де нерідко , разом з розповідями про поїздки , можна знайти оцінку практичної діяльності тих чи інших турфірм і їх партнерів. На сайті регулярно публікуються анонси статей з професійної туристичної преси. А кількість розповідей про подорожі вимірюється сотнями.

Головним недоліком даного сайту є те, що його навігаційна система не зовсім зрозуміла для користувача. Інформація на сайті неповна і не можна здійснити замовлення путівки. На даному незручно переглядати сторінки, що містять велику кількість текстової інформації - доводиться прокручувати сторінку. Так само я вважаю недоліком стандартне оформлення сайту і неоригінальний дизайн.

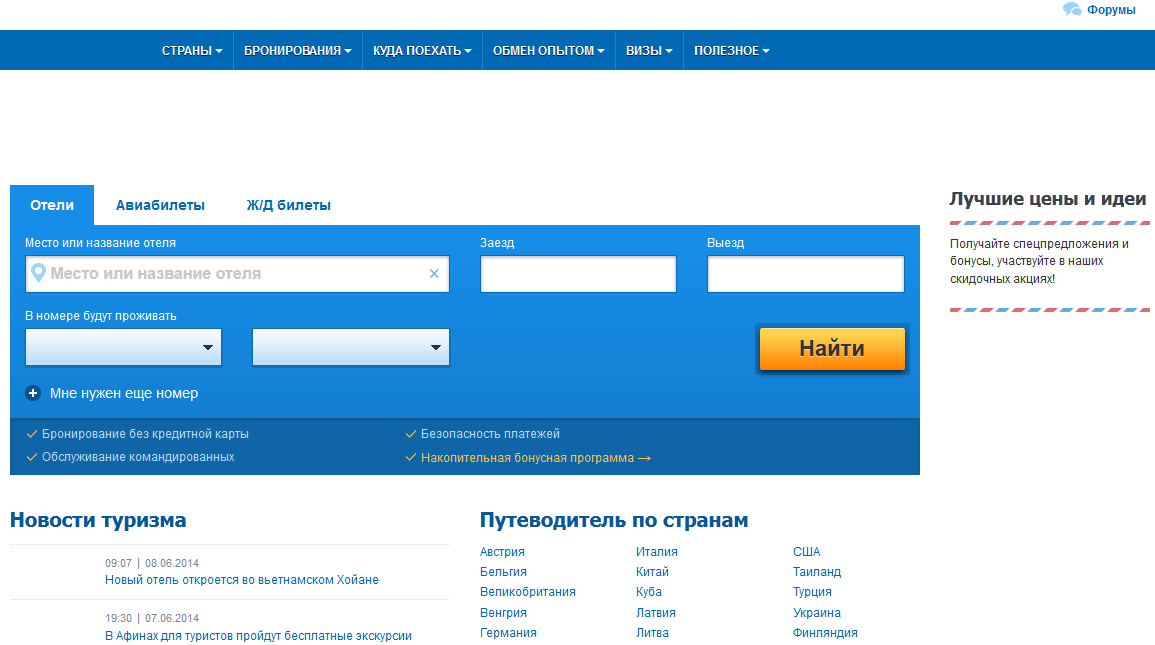


Рис.1.2 - Головна сторінка веб-сайту TRAVEL.RU

Основні можливості сайту:

* Пошук турів
* Каталог готелів
* особистий кабінет на сайті.
* Відгуки про тур
* Онлайн чат
* Відгуки про сайт
* Підписка на розсилку пропозицій
* Пошук авіаквитків на сайті

# *1.3.2.Огляд системи Art-travel*

Туристичний сервер Art-Travel є одним з великих туристичний сайтів з широким вибором туристичних послуг в декількох країнах. Він пропонує багато функціональних особливостей ,яких вимагає сучасний турист.

Основні можливості сайту:

* Пошук турів та каталог турів
* Каталог готелів по всіх популярних країнам
* Додаток для ВКонтакте і Facebook ,в яких можна знайти всі тури ,пошук авіаквитків , який включає пропозиції всіх авіакомпаній світу
* Можливість залишити заявку на довільний тур за допомогою спеціальної форми на сайті і відправити її менеджерам турагентства.
* Можливість замовити тур прямо зі сторінки туру , натиснувши на посилання « Замовити тур»
* Підписка на розсилку дозволить туристу бути в курсі свіжих новин і оновлень , при цьому турист сам обирає на яку розсилку підписатися , « гарячі тури » або новини вашої компанії або інша розсилка .



Рис.2 - Головна сторінка веб-сайту web4travel.net

* Он- лайн чат ( он- лайн консультант ) працює як Skype або ICQ , але не вимагає установки ніяких програм , відкривається натисканням кнопки на вашому сайті .
* Переклад сайту на будь-яку з 52 мов.
* Відгуки про тур. Можливість прочитати відгуки про вподобаних турах і залишити свої відгуки .
* Відгуки про сайт. Щоб залишити відгук про сайт або компанії в цілому потрібно клікнути по кнопці праворуч «Залишити відгук про сайт»
* Версія для друку для кожної сторінки дозволяє легко роздрукувати потрібний матеріал.
* Кнопки « Розмір шрифту » дозволяють збільшити / зменшити розмір шрифту.
* Модуль пошуку для сайтів партнерів дозволяє агентствам розміщувати на своїх сайтах модуль пошуку по турам
* Особистий кабінет на сайті.
* Перегляд сторінок і розділів прихованих від звичайних відвідувачів
* Можливість залишити заявку на довільний тур за допомогою спеціальної форми на сайті і відправити її менеджерам турагентства
* Підписка на розсилку пропозицій з можливістю вибрати що саме потрібно отримувати , наприклад пропозиції тільки по певній країні.
* Онлайн чат ( онлайн консультант ) працює як Skype або ICQ , але не вимагає установки ніяких програм , відкривається натисканням кнопки на сайті.
* Каталог турів наповнюється менеджерами через простий і зручний інтерфейс , що не вимагає спеціальних знань. Додавання туру займає пару хвилин. До кожного туру крім тексту таблиць і картинок можна додати прикріплений файл ( анкета або список документів наприклад) , відеоролик та фотографії.
* Модуль пошуку для сайтів партнерів працює ( наповнюється ) автоматично . Пропонуючи такий модуль агентствам партнерам ви збільшуєте їх лояльність до вашої компанії , робите співпрацю з вами більш зручним .
* Додаток для ВКонтакте і для Facebook , в яких туристи можуть шукати ваші тури , наповнюються автоматично , досить лишень додати тур на сайт і він відразу ж з'явиться в додатках. Додатки дозволяють розширити аудиторію клієнтів турпоератора за рахунок користувачів соціальних мереж .
* Пошук авіаквитків на сайті дозволяє туроператору надавати всі послуги авіакас , при цьому не вимагає додаткових витрат .
* Он-лайн заявка та чат спрощують спілкування клієнта з менеджерами , що сприяє збільшенню лояльності до компанії.
* Система розсилок дозволяє створювати кілька списків розсилки і розсилати відвідувачам саме ту інформацію , в якій вони зацікавлені .

Сайт, розроблений в ході дипломної роботи відрізняється від даних сайтів оригінальним дизайном, повнотою інформації і зручним навігаційним меню , а також великою кількістю фукнціональних особливостей ,таких як Google Maps,Panoramio.

Отже в цьому розділі було описано актуальності створення системи керування контентом дайджествого ресурсу туроператора для власника вликого туроператора. Також було розглянуто засоби реалізації системи керування контентом дайджествого ресурсу туроператора ,як клієнтської так і серверної частини. Оглянуто і проаналізовано подібні програмні рішення ,їх функціональні особливості.

# 2.ОПИС СЕРВЕРНОЇ ФУНКЦІОНАЛЬНОСТІ СИСТЕМИ КЕРУВАННЯ МЕТАКОНТЕНТОМ ДАЙДЖЕСТОВОГО РЕСУРСУ ТУРОПЕРАТОРА

# *2.1 Опис функціональних можливостей та структури бази даних системи керування ментаконтентом дайджестового ресурсу туроператора*

Структуру даних систему керування метаконтентом туроператора повина містити такі сутності:

* Тур
* Дисконтна програма
* Замовлення тура
* Користувач
* Компанія

**Тур**

Одним із важливих цілей сайту є продаж турів. Відповідно потрібно розробити структуру даних ,яка б могла надати користувачу вичерпну інформацію ,які товари і послуги надає тур ,в тому числі харчування ,проживання та інші, в яких умовах буде проживати турист, як буде здійснюватись перевезення людей . З іншого боку потрібно правильно організувати взаємозв`язки з іншими сутностями ,щоб забезпечити правильне функціонування всієї системи .

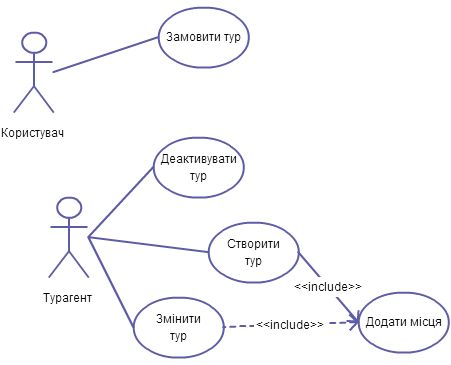


Рис.2.1 – Діаграма прецедентів туру

Як ми можемо побачити з діаграми користувач основною функцією даної частини є можливість користувача замовити тур .Турагент може створювати ,змінювати ,і деактивувати тур. Також при змінні або створення важливо ,щоб була можливість додати місця з їхніми деталями і цінами.  
 Ми бачимо тур агент не може видалити тур ,а лише деактивувати. Це пов’язано з тим ,що замовлення та інші частини системи мають зв’язок з туром і відповідно ,якщо ми видалимо тур то ця інформація зникне для користувача .

З структури БД можна виділити такі об’єкти-сутності нашої системи:

1. Загальний тура(tours).В шаблоні розміщена основна інформація про тур.
2. Конкретний тур (tour\_info).В цій таблиці розміщені початкова і кінцева дати ,а також знижка на тур.
3. Фотографії(photos).
4. Місця(places).
5. Зв'язок між турами і місцями(tours\_places).
6. Послуги ,які буде включати ціна на тур(price\_includes).
7. Зв’зок між послугами ,які включені на тур і самим туром(tours\_price\_includes).

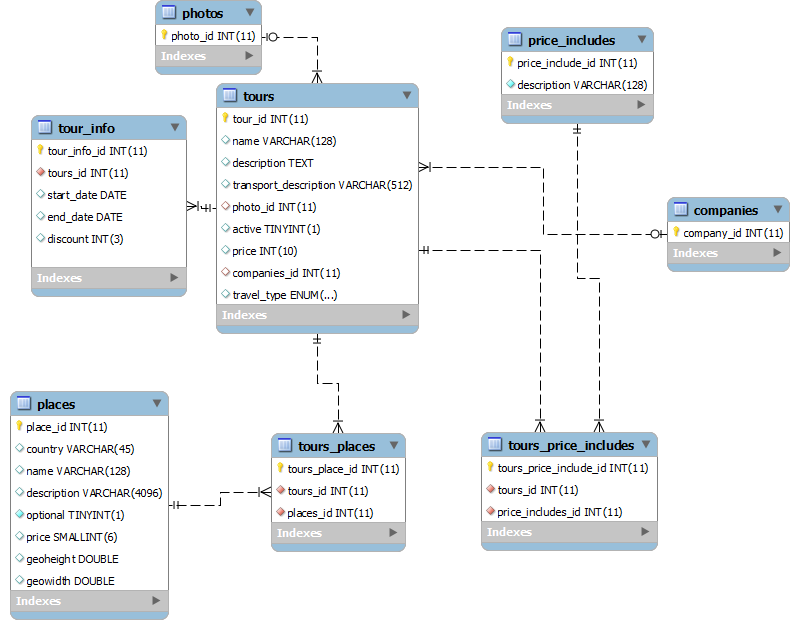


Рис.2.2 . Структура БД пов'язана з турами

**Дисконтні програми:**

В даній системі було запропоновано 3 типи знижок:

* знижка на тур
* знижка компанії
* дисконтні програми

Для функціонування дисконтних програм було запропоновано два типи дисконтів:  
а)Фіксований . Наприклад 3%

б)Динамічний . Записується ,як формула і може залежати від певних характеристик:

* кількості замовлень в історії користувача
* кількості замовлень в історії користувача на певний тур
* віку користувача
* кількості дітей або дорослих в замовленні
* наявність відгуків ,які залишав користувач
* кількості відгуків користувача про тури певної компанії

Тільки дисконтні програми можуть мати динамічну знижку.

Також дисконтні програми повинні мати умову ,яка б давала можливість створювати знижку ,яка б залежала від:

* дати . Знижки на свята ,знижка на день народження
* роду користувача
* кількості замовлень . Знижка для новачків
* кількості замовлень на певний тур
* наявності відгуку користувача про тури
* віку користувача

Як бачимо з діаграми(рис.2.3) турагент може CRUD операції з дисконтними програмами і відповідно створювати або зруйнувати зв'язок між дисконтними програмами, які він може редагувати.   
 З іншого боку адміністратор може змінювати знижку компанії ,а також робити CRUD операції з дисконтними залежностями. Дисконтні залежності створені виключно з метою зробити можливим використання   
динамічних знижок і умов для дисконтних програм.  
 При використанні дисконтних програм має бути розроблений зручний та інутїтивний інтерфейс ,який би давав можливість турагнетам швидко і легко створювати дисконтні програми , а користувачам прозоро і зрозуміло продивитись ,які можливі і вже поставленні дисконтні програми вони можуть отримати на певний тур.

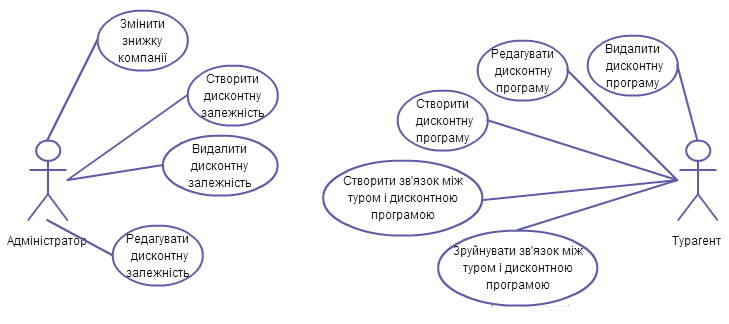


Рис.2.3 - Діаграма прецедентів дисконтних програм

З структури БД (рис.2.4) можна виділити такі об’єкти-сутності нашої системи:

1. Дисконтна програма (Discount policy)
2. Зв'язок туру і дисконтної програми(tour\_discount)
3. Дисконтні залежності(discount\_dependency).
4. Тури(tour).
5. Компанія(company).

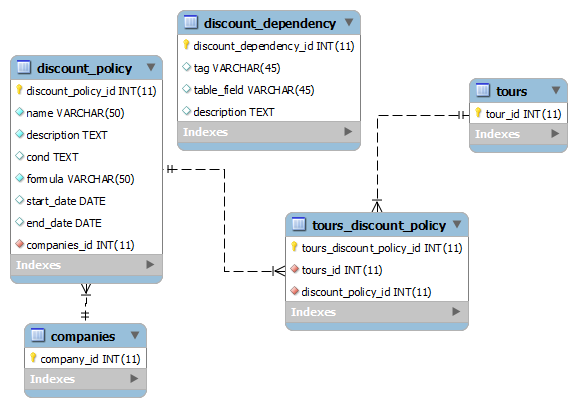


Рис.2.4 - Структура БД для дисконтних програм

**Замовлення туру**

Замовлення туру та інформація про замовлення є важливою ,як для користувача так і для турагента. В цій частині важливо щоб користувач легко і швидко міг здійснити замовлення туру , а турагент зміг знайти замовлення користувача ,зв`язатись з ним і відповідно змінювати замовлення залежно від дій користувача.

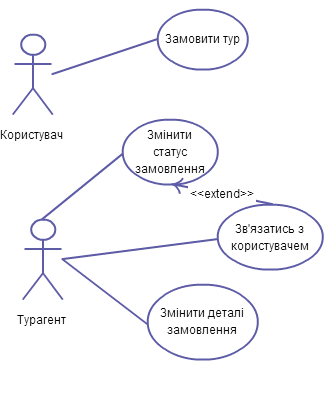
****

Рис.2.5 –Діаграма прецедентів замовлення турів

Як бачимо з діаграми спочатку користувач замовляє тур. Далі турагент зв'язується з користувачем ,щоб підтвердити інформацію або змінити деталі замовлення. Після цього турагент може змінити статус замовлення або відмінити його .

Ця система не передбачає бронювання квитків ,тому користувачу потрібно з'явитись в офіс туроператора ,щоб заповнити додаткові деталі і продовжити процес. Після успішного замовлення туру користувач може поставити рейтинг даному туру.

З структури БД(рис.2.6) замовлень можна виділити такі сутності :

1. Замовлення(orders).
2. Компанія(companies).
3. Конкретний тур(tour\_info).
4. Користувачі(users).

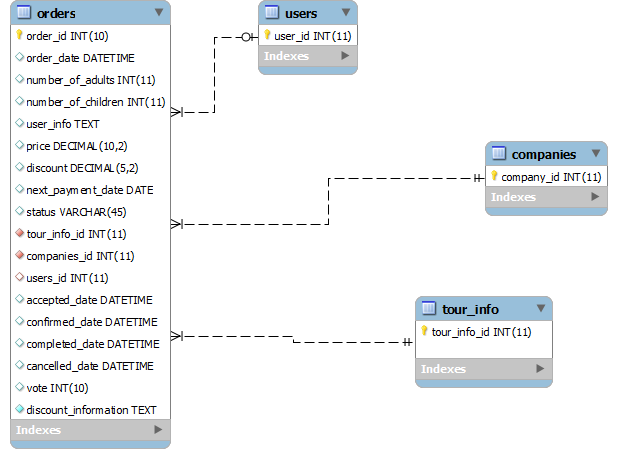
****

Рис.2.6 – Структура БД замовлень

**Користувачі**

Як можна побачити з діаграма користувачів(рис.2.7) користувач може зареєструватись ,якщо це незареєстрований користувач. Зареєстрований користувач може зайти під своїм аккаунтом, змінити деталі свого профілю , а також вийти з акаунту. Відповідно адміністратор може створвати, видаляти користувачів, а також змінювати їх профілі.

Як можна побачити з діаграма прецедентів користувачів  
три актори пов`язані з цією функціональністю. Адміністратор може створювати ,редагувати профілі та видаляти користувачів . Всі інші функції такі ж ,як і на будь-якому веб-сайті.

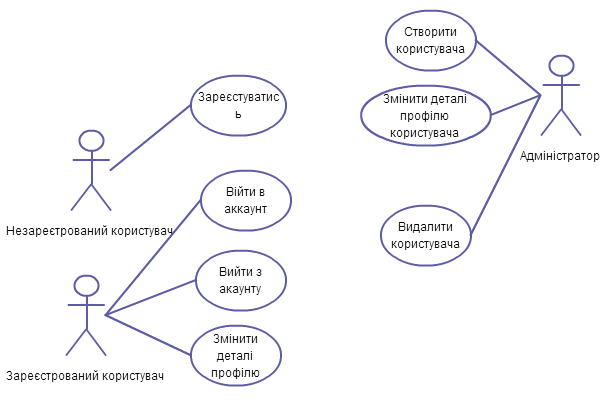


Рис.2.7 – Діаграма прецедентів користувачів

Відповідно з цього ми можемо створити наступну структуру БД.

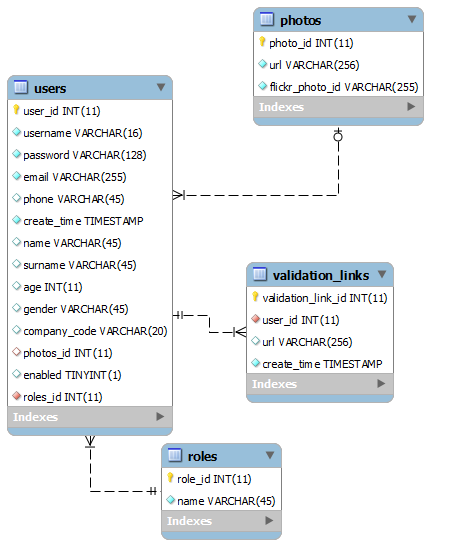
****

Рис.2.8 – Структура БД пов`язана з користувачами

Як ми бачимо можна виділити сутності:

1. Користувачі(users)
2. Ролі(roles)
3. Посилання для перевірки електроної скриньки(validation\_links)
4. Фотографії (photos)

**Компанії**

При створенні системи керування метаконтентом туроператора нам необхідно створити структуру даних ,яка б забезпечувала функціонування різних тур фірм ,які входять в склад туроператора.

****

Рис.2.9 – Діаграма прецедентів компаній

Для цього ми розглянемо які функції повинні виконувати різні актори нашої системи. Як ми бачимо з діаграми тільки адміністратор може створювати, змінювати деталі компанії і видаляти компанії. Звичайні користувачі можуть лише вступити в компанію або покинути її.

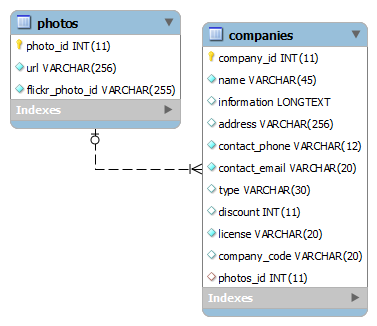


Рис. 2.10 – Структура БД компанії

# 3.РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КЕРУВАННЯ МЕТАКОНТЕНТОМ ДАЙДЖЕСТОВОГО РЕСУРСУ ТУРОПЕРАТОРА

# *3.1 Опис програмного продукту*

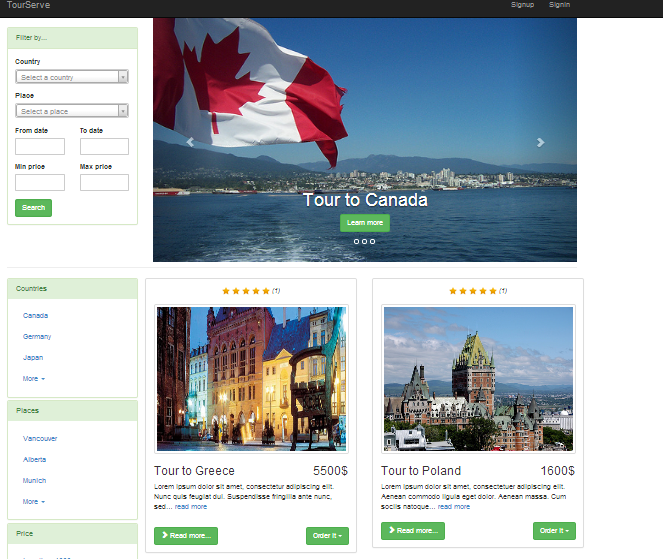
На головному вікні веб-сайту(рис.3.1) ми можемо побачити ,що для зручності користування і швидкій навігації дана сторінка розбита на блоки,які ми можемо побачити зверху. Перший елемент сторінки - навігаційне меню веб-сайту ,яке дає можливість перейти на головну сторінку , зареєструватись і авторизуватись ,якщо потрібно користувачу. Зліва користувач може задати по який критеріями він хоче здійснити пошук турів. Посередині на веб-сайті розміщено 3 найпопулярніших тури . Далі нижче зліва ми можемо побачити по який країнам і місця користувач може фільтрувати тури. І останній елемент цією сторінки опис самих турів,який складається з рейтингу туру,якщо він є, фото туру, короткого опису,ціни і можливості переглянути або замовити тур.  


Рис.3.1 - Головне вікно веб-сайту

Далі якщо користувач захоче переглянути деталі туру він може вибрати потрібній тур і натиснути на посилання “Read more”. Також користувач може замовити тур натиснувши на посилання “Order it”

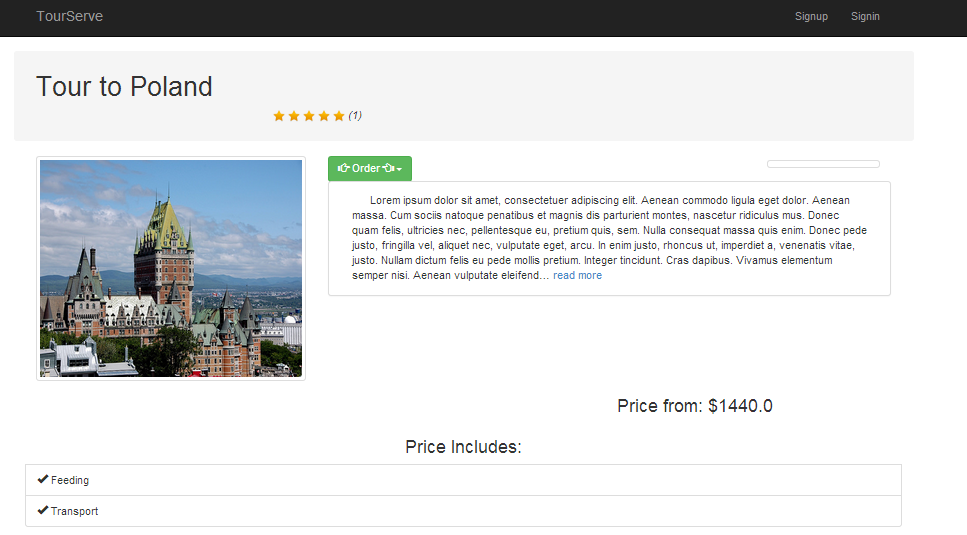


Рис. 3.2 – Вікно деталей туру

На цьому вікні ми можемо побачити рейтинг туру , посилання для вибору конкретного туру і його замовлення “Order it”,зліва знаходиться фотографія туру,яку вибрав тур агент при створенні туру,коротки опис туру ,який користувач може розгорнути ,якщо натисне на посилання “read more”. Далі користувачу відображається ціна туру і послуги , які включає ціна на тур.

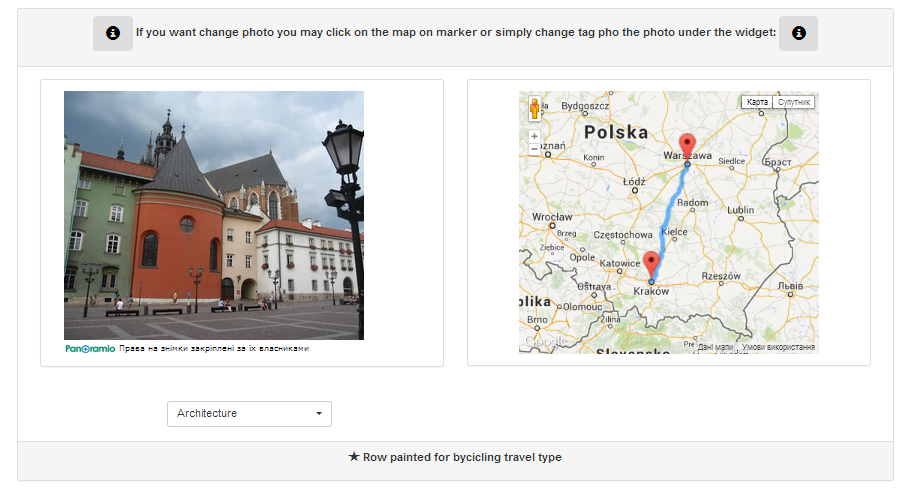


Рис. 3.3 – Google maps i Panoramio

На рис.3.3 ми можемо побачити продовження опису туру і тут є дві основні функціональні особливості ,які відрізняють програмний продукт на фоні інших подібних систем:

* Google maps за допомогою якого користувач може побачити маршрут по якому буде вібуватись тур
* Panoramio – віджет який дозволяє користувача подивитсь фотографії кінцевої зупинки туру. Також користувач може вибрати категорію , щоб фільтрувати фотографії.

Далі ми розглянемо реєстрації користувача.

За допомогою Twitter Bootstrap форма реєстрації користувача(рис.13) має приємний вигляд і дає користувачу можливість вказати всі необхідні дані для реєстрації. Також він може вибрати свою фотографію і вступити в компанії ,якщо правильно вкаже company code .

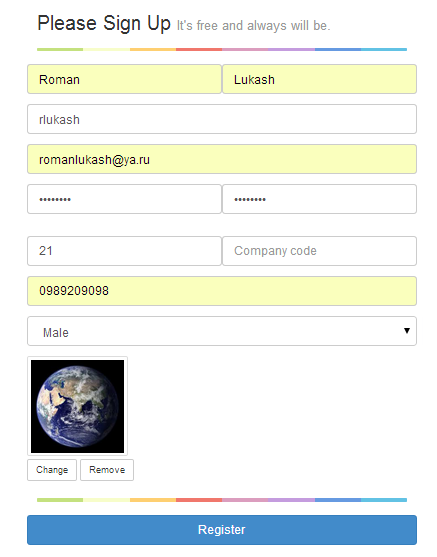


Рис.3.4 – Форма реєстрації

Якщо валідація форми пройде успішно він побачить повідомлення з подальшими інструкціями(рис.3.5) ,щоб активувати аккаунт.

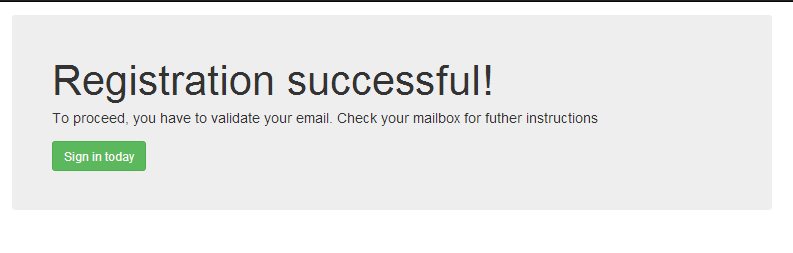


Рис.3.6 – Повідомлення про закінчення реєстрації

Далі система створює посилання для активації аккаунта ,яке зберігається в БД і є активним лише 2 дні . Для підтвердження власності електоної пошти, яка була вказана при реєстрації користувач повинен зайти на свою електрону пошту де він побачить лист надісланий від імені компанії з посиланням на активацію аккаунта.

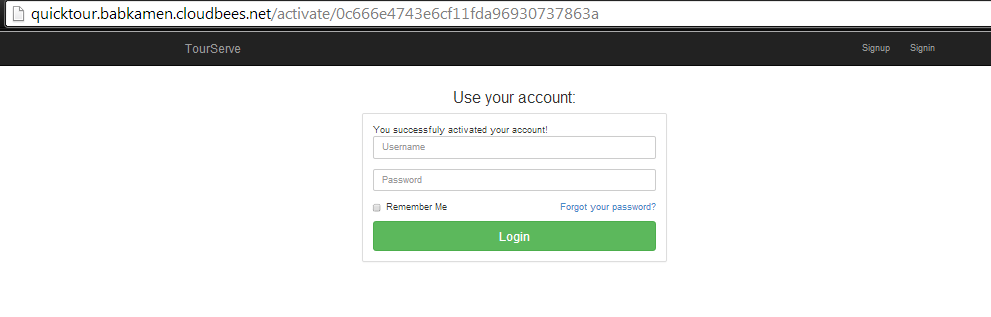


Рис.3.7 – Форма входу в аккаунт після активації

Як ми бачимо з рис.3.7 користувач активував свій аккаунт і тепер може на нього зайти. Якщо користувач зробив помилку при реєстрації він може змінити деталі свого профілю якщо з будь-якої сторінки сайту перейде по посиланню “Profile” в вкладці з назвою його аккаунту.

Як ми бачимо з рис.3.8 користувач може змінити більшість даних його профілю . Також як і на сторінці реєстрації при зміні фото користувач може зразу побачити свій можливий аватар.

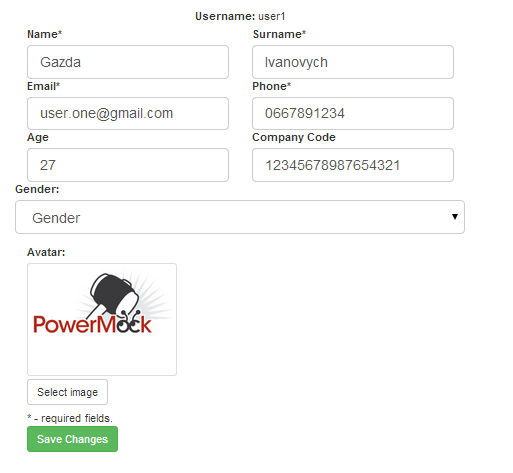


Рис.3.8 – Профіль користувача

Якщо користувач вступи в компанію він може побачити деталі компанії ,якщо перейде по посиланню “company” в вкладці назва свого аккаунта.З рис.3.9 можна побачити ,що користувач може побачити всі деталі компанії ,знижку компанії ,а також при потребі змінити компанію за допомогою коду компанії.

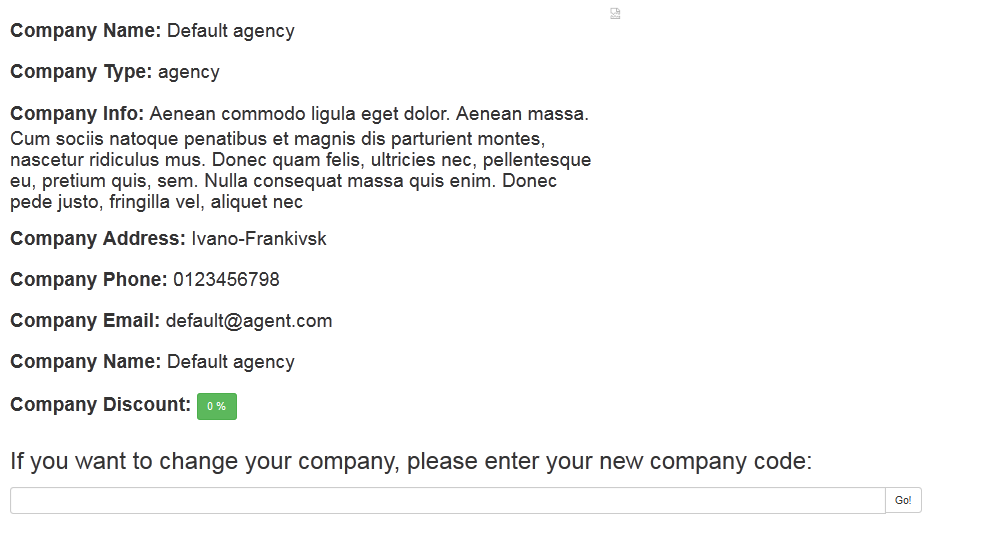


Рис.3.9 – Сторінка компанії користувача

Після того ,як користувач замовить тур з сторінки деталей туру або з головної сторінки він потрапить на сторінку замовлення туру(рис.3.10). На цій сторінці користувач може побачити деталі туру, ціну туру з знижками, а також блок для зміни особистої інформації . Далі він може побачити список знижок ,а також вказати кількість дорослих і дітей в замовленні . Також користувач може перерахувати знижки ,якщо він змінив деталі замовлення. При потребі користувач може залишити коментар до замовлення. Користувач може підтвердити замовлення туру натиснувши на посилання “Сonfirm order ”, або відмінити замовлення , натиснувши на посилання “Cancel ”.

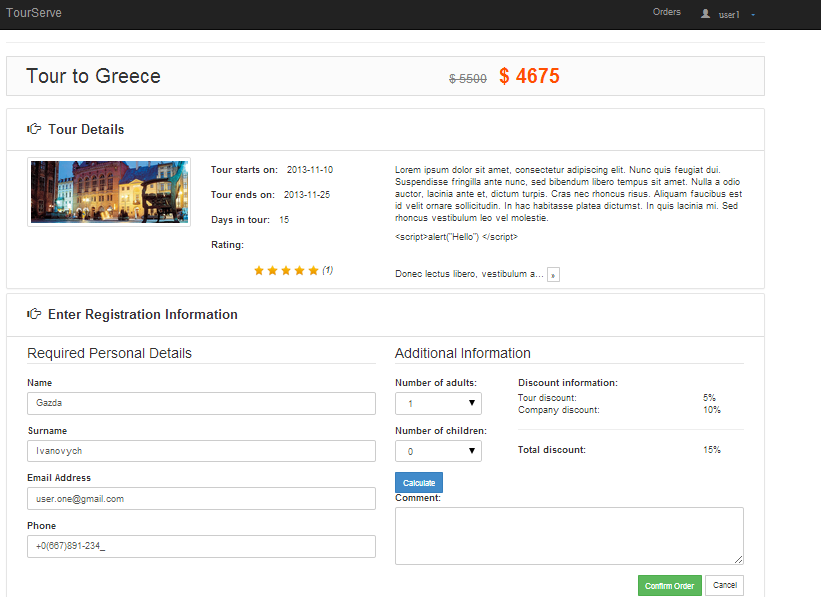


Рис.3.10 – Замовлення туру

В навігаційній панелі користувач може подивися історію своїх замовлень. На ній користувач може побачити назву туру, рейтинг туру, ціну туру, дату замовлення ,статус замовлення, наступна дата оплати, і кнопку для зміни замовлення.

Для зручності на сторінці можна фільтрувати замовлення по статусу , а також на сторінці створена пагінація для зручного відображення замовлень.

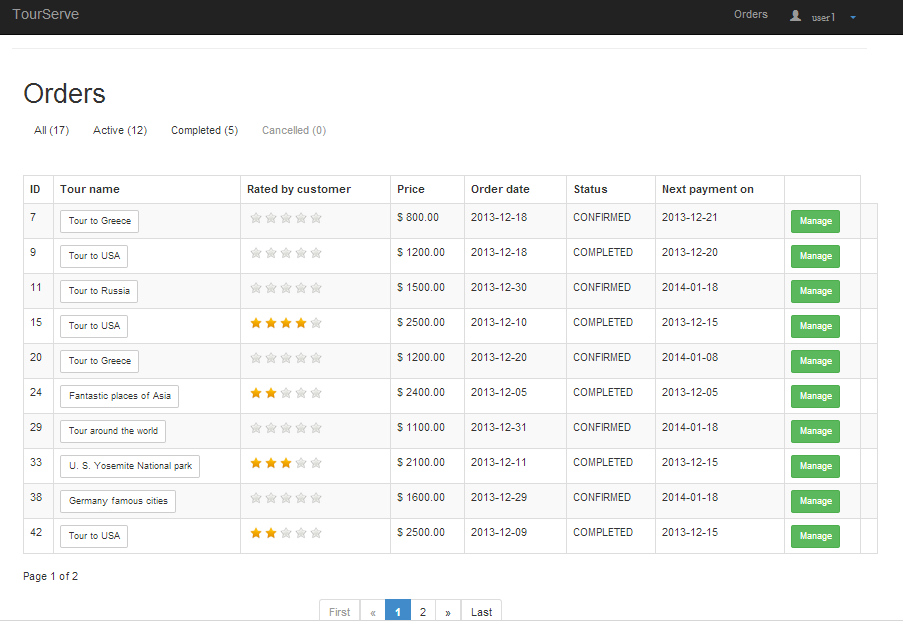


Рис.3.11 – Список замовлень

А тепер ми виконаємо дії для відновлення втрачено паролю. Спочатку користувач має натиснути посилання “Forgot password” на формі входу(рис.3.7).Після цього користувач має вести свій email(рис.3.12).

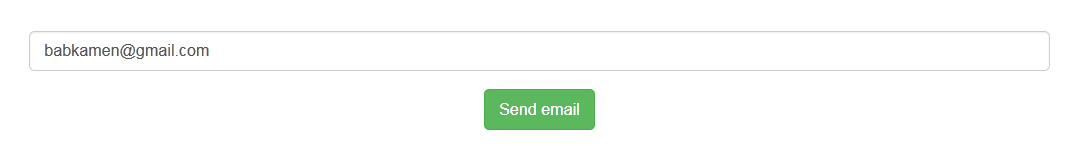


Рис.3.12 – Форма відновлення паролю

Якщо користувач введе правильний email то система створить коротке посилання і надішле його на електрону пошту користувача(рис.3.13).

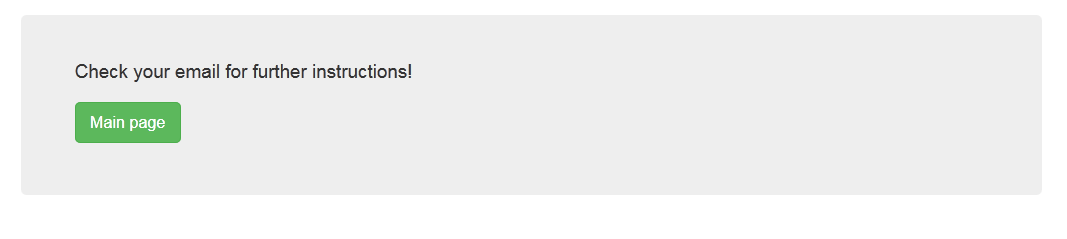


Рис.3.13 – Повідомлення про подальші інструкції щодо відновлення паролю

Після цього користувач мусить зайти на свою електрону пошту і перейти по створеному посиланню. Після цього він може змінити пароль до свого аккаунту (рис.3.14).

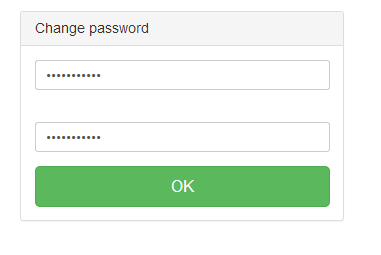


Рис. 3.14 – Зміна паролю при його втраті

Якщо користувач введе пароль достатньої довжини і обидва паролі співпадають система змінить пароль. Після цього користувач побачить повідомлення ,що система змінила пароль(рис.3.15).

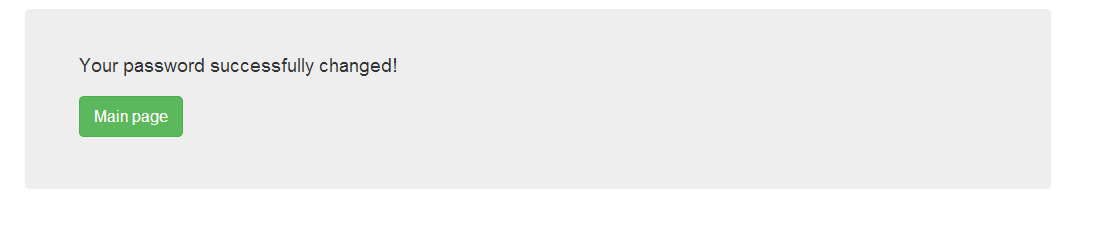


Рис.3.15 – Повідомлення про успішну зміну паролю

Якщо ми користувач має роль адміністратора , і натисне на посилання “Users ”в вкладці “View all” ,то він побачить список всіх користувачів на веб-сайті(рис3.16). На цій сторінці адміністратор може побачити деталі профілю користувачів, їхню роль, а також має змогу редагувати і видалити користувачів.

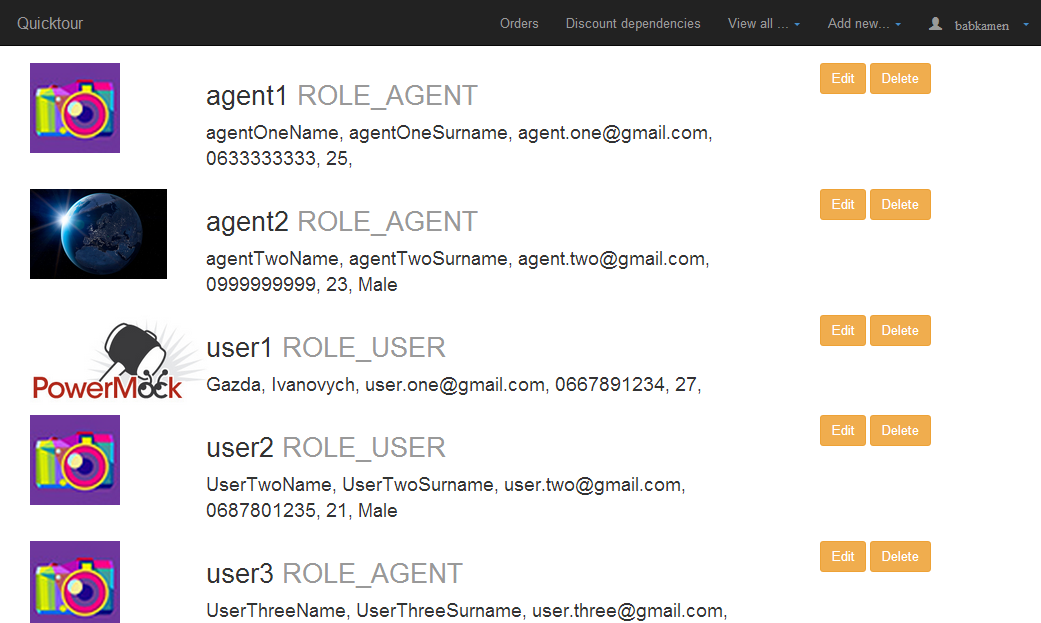


Рис.3.16- Список користувачів

Якщо адміністратор натисне на посилання “Companies” в вкладці “View all..” він може побачити список компаній на веб-сайті. На сторінці списку компаній адміністратор може побачити фотографію компанії, її назву ,тип компанії , а також можливість видалити ,змінити деталі компанії , а також створити нову компанію.

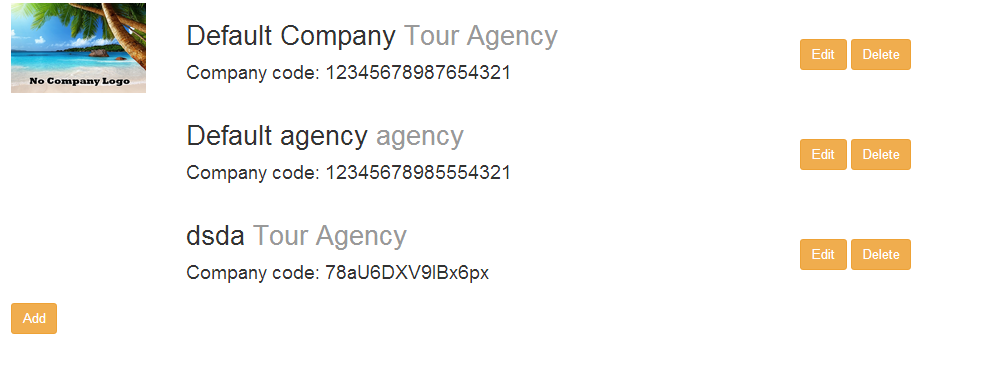


Рис.3.17 - Список компаній

Якщо натисне на посилання “Edit” то він перейде на сторінку редагування компанії.

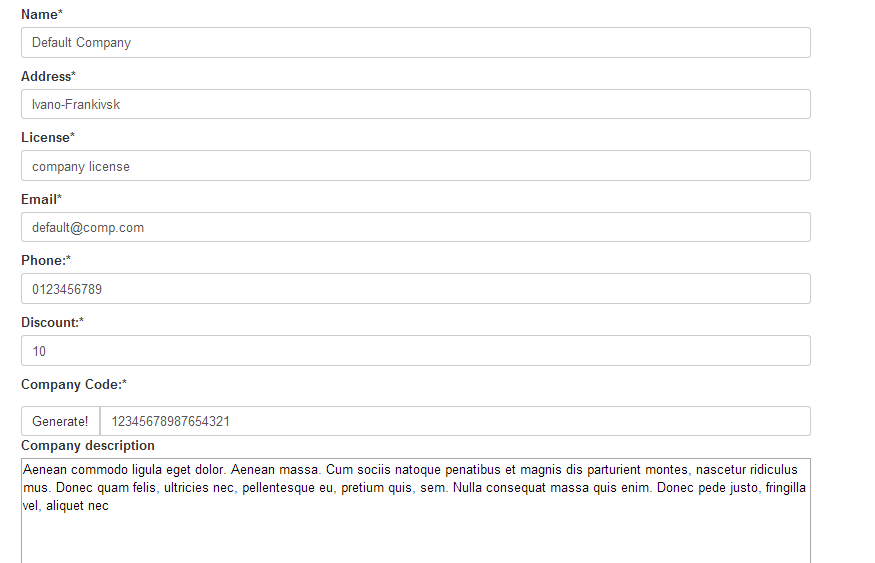


Рис.3.18 – Форма реєстрації або зміни компанії

На цій сторінці адміністратор може змінити деталі компанії : назву, адресу,поштову скриньку , знижку компанії, опис компанії , а також змінити або згенерувати новий код компанії по якому користувач може вступити в компанію.

Якщо адміністратор натисне на посилання “Discount dependencies” в навігаційній панелі ,то він зможе побачити список дисконтних залежностей.

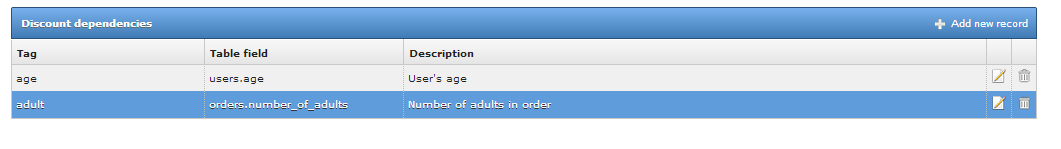


Рис. 3.19 – Сисок дисконтних залежностей

На ньому він може робити виконати CRUD операції пов`язані з дисконтними залежностями . Якщо адміністратор натисне на посилання “Add new record” ,то відкриється діалогове вікно.

На цьому кроці адміністратор може заповнити всі потрібні дані ,а також для зручності вибрати з випадючого списку потібні дані. Після цього відбудеться валідація форми і система додасть новий елемент в БД.

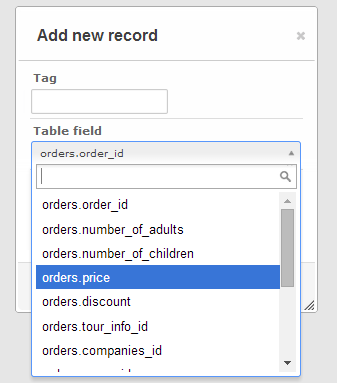


Рис.3.20 – Додавання нової дисконтної залежності

Для турагента самих головнішим на сайті є можливості робити CRUD операції з турами.Для відображення списку турів турагенту потрібно натиснути на посилання “Tours” в вкладці “Manage tours”.

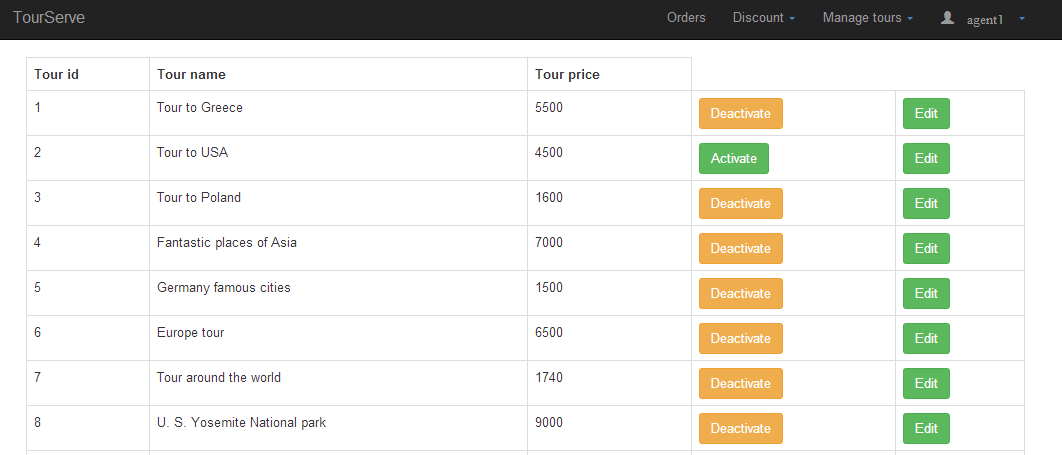
Турагент на цьому списку може побачити id, назву ,ціну туру . Він має можливість деактивувати тур , тоді користувачі не зможуть побачити тур на веб-сайті. Також в нього є можливість змінити деталі вибраного туру.  
  


Рис.3.21 – Список турів

Для турагента є важливим мати можливість просто і швидко створювати тур. Для того щоб створити тур йому потрібно натиснути на вкладку “Сreate new tour” в вкладці “Manage tours”.

На сторінці створення туру(рис.3.22) тур агент має вказати назву туру, чи активний це буде тур після створення , опис туру в редакторі,фотографію і опис туру, вид транспорту ,а також опис поїздки ,послуги які включає в себе ціна

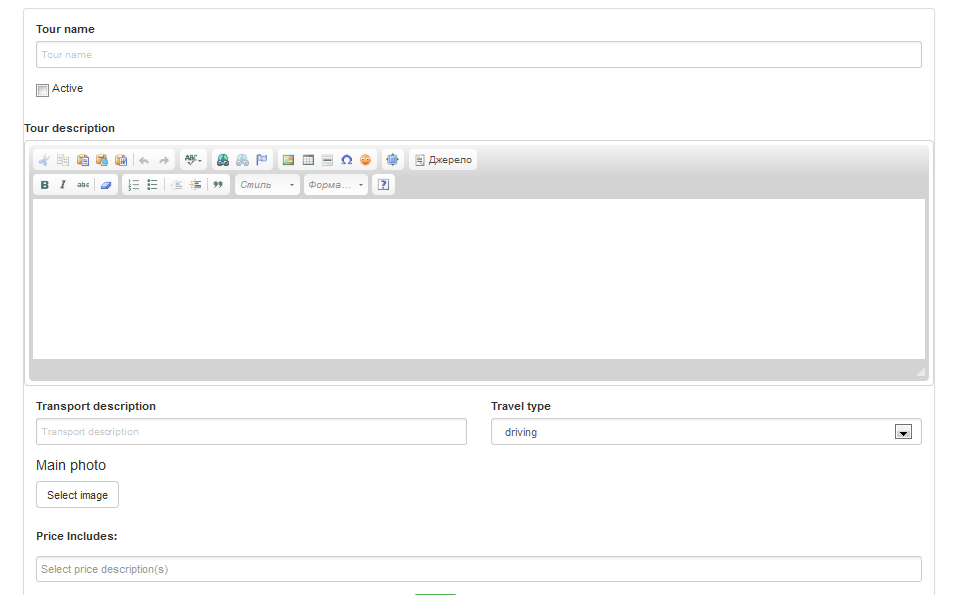


Рис.3.23 – Вікно додавання туру

Після того як система перевірить чи все заповнено вірно ,турагент може перейти на настуний крок (рис.3.24),де він має вказати дати турів, а також знижку на ці конкретні тури. Для зручності за допомогою Javascript можна додавати і видаляти вікна для заповнення певного конкретного туру.

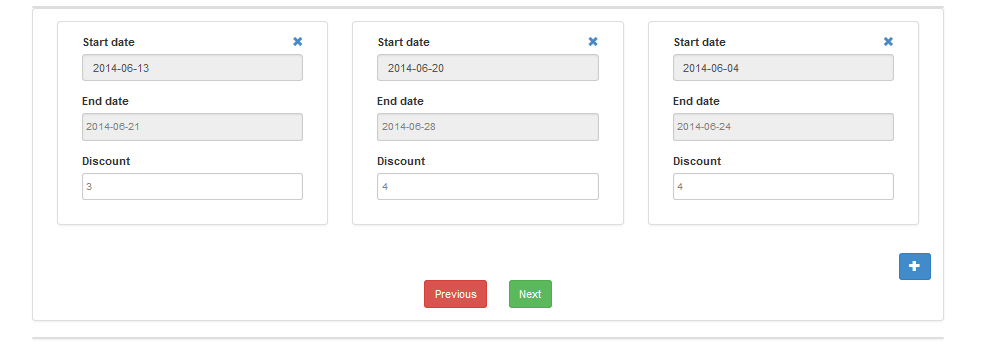


Рис.3.24 Вибір дат при додаванні туру

Якщо тур агент ввів неправильні дані на попередньому кроці , він може повернутись натиснувши на посилання “Previous” і відповідно перейти на наступний крок натиснувши на посилання “Next”.

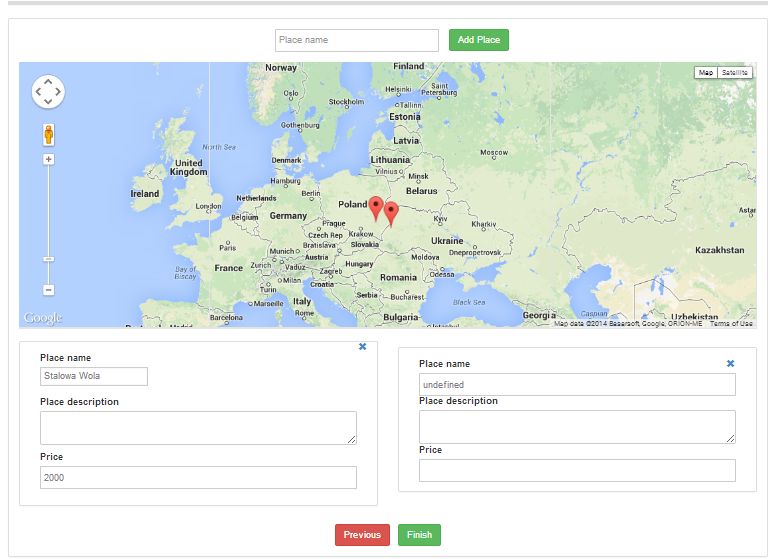


Рис.3.25 – Вибір місць при додаванні туру

В останньому кроці турагент має вказати маршрут по якому буде відбуватись тур. Для цього за допомогою Google maps він може створити маркеру ,а потім система сама прокладе маршрут. Щоб додати місця турагент може вказати назву місця і натиснути на посилання “Add place і якщо таке місце існує в сховищі Google Maps, то створиться нове вікно ,де буде назва місця . Турагенту потрібно вказати ціну місця ,і додатково опис ,але це не є обов`язковим. Після цього турагент може завершити створення туру натиснувши на посилання “Finish”.  
 Отже в даному розділі було оглянуто і проаналізовано головні функціональні особливості програмного продукту . Було оглянуто основні елементи і їх властивості користувацького інтерфейсу.

# 4.ОХОРОНА ПРАЦІ

# *4.1.Законодавче та нормативно-правове забезпечення охорони праці*

Основою законодавства України з охорони праці є Конституція України, що гарантує громадянам право на безпечні й здорові умови праці й система законодавчих актів України, спрямованих на реалізацію цього конституційного права[17].

Основними законодавчими актами цієї системи є наступні Закони України:

• «Про охорону праці».

• «Про охорону здоров'я».

• «Про пожежну безпеку».

• «Про обов'язкове державне соціальне страхування від нещасного випадку

на виробництві й професійного захворювання, що привели до втрати працездатності».

• «Про використання ядерної енергії і радіаційний захист».

• «Про забезпечення санітарного й епідемічного благополуччя населення».

• «Про цивільну оборону».

• Кодекс законів «Про працю України».

У наведеній вище системі законодавчих актів основна роль приділяється

Закону «Про охорону праці» (редакція від 2002 р.). Цей закон визначає основні положення з реалізації конституційного права громадян на охорону їхнього життя й здоров'я у процесі трудової діяльності, регулює за участю відповідних державних органів відносини між власником підприємства і працівником з питань безпеки праці, виробничої санітарії, встановлює єдиний порядок організації охорони праці у виробничій сфері в Україні. Чинність Закону України «Про охорону праці» поширюється на всі підприємства, установи, організації незалежно від форм власності й видів діяльності, що використовують найману працю, і на всіх працюючих.

Закон визначає основні принципи державної політики в області охорони

праці, серед яких чільне місце займають:

• пріоритет життя й здоров'я працівників стосовно результатів виробничої

діяльності підприємства;

• повна відповідальність власника підприємства за створення безпечних і нешкідливих умов праці;

• соціальний захист працівників;

• повне відшкодування шкоди особам, які потерпіли на виробництві від

нещасних випадків або професійних захворювань.

Окремо виділені статті Закону присвячені регулюванню охорони праці

жінок, неповнолітніх, інвалідів, видам відповідальності за порушення законодавства і нормативних актів про охорону праці, за створення перешкод для діяльності посадових осіб органів державного нагляду за охороною праці і представників профспілок.

# *4.2.Організація роботи з охорони праці на підприємстві*

Колективний договір є найбільш важливим документом у системі нормативного регулювання взаємин між власником підприємства і працівниками з першочергових соціальних питань, у тому числі з питань охорони праці. Питанням охорони праці присвячений спеціальний розділ колективного договору, що називається «Охорона праці»[18].

Положенням, що викладаються в колективному договорі, повинні передувати колективні переговори, що починаються сторонами за три місяці до закінчення терміну дії попереднього договору. Зобов'язання, які пропонується включити в колективний договір за результатами переговорів, повинні бути реальними й всебічно обґрунтованими, тому що після схвалення і підписання документа вони стають обов'язковою нормою для виконання, яка діє в рамках підприємства.

Зобов'язання, викладені в колективному договорі (далі Договорі) не повинні суперечити законам й нормативним актам України. Гарантії, пільги і компенсації, наведені в Договорі згідно з чинним законодавством, вважаються обов'язковими для виконання при будь-яких обставинах.

Колективний договір повинен обов'язково містити заходи щодо захисту прав і соціальних інтересів осіб, які постраждали на виробництві від нещасних випадків, а також їх утриманців і членів родин загиблих. Вони повинні включати такі положення:

• з відшкодування нанесеного збитку здоров'ю потерпілим;

• про виплату одноразової допомоги, компенсації витрат на придбання медикаментів, на додаткове оплачуване лікування;

• про надання потерпілим, відповідно до медичного висновку, більш легкої

роботи, при одночасному збереженні середнього заробітку;

• про організацію навчання, перекваліфікації потерпілих, працевлаштування інвалідів праці, наданню таким інвалідам допомоги у вирішенні соціально-побутових питань і т. п.

Усі працівники, відповідно до законодавства, підлягають обов'язковому страхуванню від нещасних випадків на виробництві і професійних захворювань, що призвели до втрати працездатності (1.7).

Згідно із ст. 20 Закону в підрозділі колективного договору про охорону

праці повинні бути обов'язково відображені наступні заходи[19]:

• забезпечення працівникам соціальних гарантій в області охорони праці на

рівні, не нижче передбаченого законодавством;

• комплексні заходи для досягнення нормативів безпеки праці й виробничої санітарії;

• заходи з підвищення існуючого рівня охорони праці, попередження випадків виробничого травматизму, професійних захворювань, аварій і пожеж.

Повинні бути також визначені обсяги і джерела фінансування вказаних заходів.

Зобов'язання угоди «Охорона праці» колективного договору повинні відповідати принципам економічного стимулювання і матеріальної відповідальності за стан охорони праці (ст. 25 Закону «Про охорону праці»). Зобов'язання колективного договору є двосторонніми, тому цей документ повинен містити не тільки вимоги до власника (адміністрації підприємства), але і зобов'язання працівників з безумовного виконання норм, правил, стандартів і інструкцій з охорони праці, дотримання встановлених вимог користування машинами, механізмами, інструментом і пристроями, обов'язкового використання засобами колективного й індивідуального захисту і т. д.

Залежно від характеру виробництва, складу трудового колективу, специфіки галузі в колективному договорі визначаються зобов'язання з організації безпечних і нешкідливих умов праці інвалідів, неповнолітніх, пенсіонерів, осіб, які тимчасово залучаються до виконання суспільних робіт за договорами з центрами зайнятості населення та ін[20]. У трудові договори забороняється включати умови, що погіршують у порівнянні з чинним законодавством, колективним договором становище працівників. Створення безпечних і здорових умов праці забезпечується шляхом впровадження системи організації охорони праці на виробничому рівні. Закон "Про охорону праці" містить окремий розділ, присвячений організації охорони праці та її управлінню на підприємствах, в установах, організаціях.

Система управ­ління охороною праці передбачає:

* створення роботодавцем відповідних служб, які забезпе­чують вирішення конкретних питань охорони праці;
* досягнення встановлених нормативів з охорони праці шляхом впровадження прогресивних технологій, додержання вимог при проектуванні, будівництві та реконструкції підпри­ємств, об'єктів і засобів виробництва;
* проведення лабораторних досліджень умов праці, атестації робочих місць на відповідність нормативним актам про охорону праці та надання пільг і компенсацій за результатами їх про­ведення;
* проведення обов'язкових медичних оглядів певних кате­горій працівників, навчання з охорони праці;
* усунення причин, що призводять до нещасних випадків, професійних захворювань, здійснення їх розслідування й обліку тощо.

На виробництві можуть впроваджуватися й інші заходи з організації охорони праці. До кола питань з управління охоро­ною праці включають: забезпечення безпеки виробничого устаткування, виробничих процесів, будівель та споруд, норма­лізацію санітарно-гігієнічних умов праці, забезпечення працюючих засобами індивідуального захисту, встановлення оптимальних режимів праці та відпочинку працюючих, сані­тарно-побутового обслуговування працюючих та їх професійний відбір за окремими спеціальностями[21].

Одним з основних суб'єктів, що здійснюють управління охороною праці на підприємстві, є служба охорони праці, ор­ганізаційна форма якої залежить від кількості працівників, що працюють у роботодавця. На підприємстві, де працюють 50 і більше осіб, роботодавець створює службу охорони праці, на підприємстві, де працює менше 50 осіб, функції служби охорони праці можуть виконувати в порядку сумісництва особи, які мають відповідну підготовку, а на підприємстві, де працює менше 20 осіб, для виконання функцій служби охорони праці можуть залучатися сторонні спеціалісти на договірних засадах, які мають відповідну підготовку. Ліквідація служби охорони праці допускається лише у разі ліквідації підприємства чи припинення фізичною особою-роботодавцем використання найманої праці.

|  |
| --- |
|  |

Ефективна організація охорони праці передбачає постійну співпрацю роботодавця з найманими працівниками та їх пред­ставниками, створення з ними спільних органів управління. До таких суб'єктів належить комісія з питань охорони праці підприємства. Комісія з питань охорони праці підприємства створюється лише за рішенням загальних зборів (конференції) найманих працівників. Вона є постійно діючим консультативно-дорадчим органом найманих працівників та роботодавця, тому на відміну від служби охорони праці її рішення мають реко­мендаційний характер.

Система управління охороною праці включає додержання вимог з охорони праці при проектуванні, будівництві (виготов­ленні) та реконструкції підприємств, об'єктів і засобів вироб­ництва, їх відповідності нормативно-правовим актам з охорони праці.

Роботодавець зобов'язаний провести (забезпечити) лаборатор­ні дослідження умов праці, де наявні шкідливі та важкі умови праці, встановити пільги і гарантії для працівників, які працю­ють у цих умовах, та розробити заходи, що забезпечують усунення причин виникнення нещасних випадків і професійних захворювань. Атестація робочих місць є комплексною оцінкою всіх факторів виробничого середовища і трудового процесу, супутніх соціально-економічних факторів, що впливають на здоров'я і працездатність у процесі трудової діяльності.

Для підприємств, незалежно від форм власності, або фізичних осіб, які використовують найману працю, витрати на охорону праці становлять не менше 0,5 відсотка від суми реалізованої продукції. На підприємствах, що утримуються за рахунок бюджету, витрати на охорону праці передбачаються в державному або місцевих бюджетах і становлять не менше 0,2 відсотка від фонду оплати праці. Суми витрат з охорони праці, що належать до валових витрат юридичної чи фізичної особи, яка відповідно до законодавства використовує найману працю, визначаються згідно з переліком заходів та засобів з охорони праці, що затверджується Кабінетом Міністрів України.

Кошти галузевих і державного фондів охорони праці витрачаються на здійснення галузевих і національних програм з питань охорони праці, науково-дослідних і проектно-конструкторських робіт, що виконуються в межах цих програм, на сприяння становленню і розвитку спеціалізованих підприємств та виробництв, творчих колективів, науково-технічних центрів, експертних груп, на заохочення трудових колективів і окремих осіб, які плідно працюють над розв'язанням проблем охорони праці.

Роботодавець зобов'язаний за свої кошти організувати проведення попереднього (при прийнятті на роботу) і періодичних (протягом трудової діяльності) медичних оглядів працівників, зайнятих на важких роботах, роботах із шкідливими чи небезпечними умовами праці або таких, де є потреба у професійному доборі, а також щорічного обов'язкового медичного огляду осіб віком до 21 року. За результатами періодичних медичних оглядів у разі потреби роботодавець повинен забезпечити проведення відповідних оздоровчих заходів. Медичні огляди проводяться відповідними закладами охорони здоров'я, працівники яких несуть відповідальність згідно із законодавством за відповідність медичного висновку фактичному стану здоров'я працівника. Проведення медичних оглядів визначається спеціально уповноваженим центральним органом виконавчої влади в галузі охорони здоров'я.

Роботодавець має право в установленому законом порядку притягнути працівника, що ухиляється від обов'язкового медичного огляду, до дисциплінарної відповідальності, а також зобов'язаний відсторонити його від роботи без збереження заробітної плати.

Роботодавець зобов'язаний за свій рахунок забезпечити позачерговий медичний огляд працівників:

* за заявою працівника, коли він вважає, що погіршення стану його здоров'я пов'язане з умовами праці;
* за своєю ініціативою, коли стан здоров'я не дозволяє працівнику виконувати свої трудові обов'язки.

За час проходження медичного огляду за працівниками зберігаються місце роботи (посада) і середній заробіток.

Навчання працівників з питань охорони праці є однією зі складових організації охорони праці. Усі працівники при прийнятті на роботу і в процесі роботи проходять інструктаж, навчання з питань охорони праці, надання першої медичної допомоги потерпілим від нещасних випадків і правил поведін­ки при виникненні аварій.

На роботодавця також покладаються обов'язки з розсліду­вання та обліку нещасних випадків і професійних захворювань, вивчення причин, що спричинили настання негативних наслідків виробничого травматизму. Процедура розслідування і обліку нещасного випадку передбачає повідомлення відповідних суб'єктів про нещасний випадок, що трапився, і надання необ­хідної допомоги потерпілим та вжиття превентивних заходів щодо попередження виникнення інших нещасних випадків; створення комісії з розслідування нещасних випадків; проведен­ня розслідування нещасних випадків, оформлення результатів розслідування та облік нещасних випадків.

Навчання з питань охорони праці. Питанням охорони праці працівники повинні навчатися постійно.

Усі працівники при прийнятті на роботу і в процесі роботи повинні проходити за рахунок роботодавця на підприємстві інструктаж, навчання з питань охорони праці, з надання першої медичної допомоги потерпілим від нещасних випадків і правил поведінки при виникненні аварії.

Працівники, зайняті на роботах з підвищеною небезпекою або там, де є потреба у професійному доборі, повинні проходити попереднє спеціальне навчання і один раз на рік перевірку знань відповідних нормативно-правових актів про охорону праці. Перелік робіт з підвищеною небезпекою затверджується спеціально уповноваженим центральним органом виконавчої влади з нагляду охороною праці.

Посадові особи, діяльність яких пов'язана з організацією безпечного ведення робіт, під час прийняття на роботу і періодично один раз на три роки, проходять навчання, а також перевірку знань з питань охорони праці за участю профспілок.

Працівники, у тому числі посадові особи, які не пройшли навчання, інструктаж і перевірку знань з охорони праці, не допускаються до роботи. Коли у працівників, у тому числі посадових осіб, виявлені незадовільні знання з питань охорони праці, вони повинні у місячний строк пройти повторне навчання і перевірку знань.

# *4.3.Основні причини профзахворювань і заходи щодо їх попередження*

Нерухома напружена поза програміста призводить до втоми і виникнення болю в хребті, шиї, плечових суглобах. Інтенсивна робота з клавіатурою викликає болючі відчуття в ліктьових суглобах, передпліччях, зап'ястях і пальцях рук[22]. У деяких операторів персонального комп'ютера (ПК) розвивається м'язова слабкість, відбувається зміна форми хребта (синдром тривалого статичного навантаження - СТСН), що може призвести до непрацездатності. Постійні користувачі ПК найчастіше піддаються психічним стресам, хворобам серцево-судинної системи і верхніх дихальних шляхів. Значному навантаженню піддається зоровий апарат.

При тривалій та інтенсивній роботі за комп'ютером з'являється синдром комп'ютерного стресу (СКС), який проявляється головною біллю, запаленням очей, алергією, дратівливістю, млявістю і депресією, погіршенням зосередженості і працездатності.

Причинами різноманітних симптомів СКС є 5 основних чинників: неправильна робота очей і поза тіла, носіння невідповідних окулярів або контактних лінз, неправильна організація робочого місця, розподілення фізичних, розумових, візуальних навантажень, низький рівень візуальної підготовленості для роботи з комп'ютером.

Інтенсивна і тривала робота з клавіатурою комп'ютера може стати джерелом важких професійних захворювань рук. Робота з клавіатурою є причиною 12% профзахворювань, викликаних повторюваними рухами.

Захворювання, пов'язані з "травмами повторюваних навантажень", пов'язані з хворобами нервів, м'язів і сухожилків . Наприклад тендит - запалення і набрякання сухожилків . Захворювання поширюється на кисті рук, зап'ястя, плечі. Інше - травматичний епікон-диліт - подразнення сухожилів, що з'єднують м'язи передпліч та ліктьових суглобів.

Хвороба Де-Кервена - різновид тендита, при якій страждають сухожилля, пов'язані з великим пальцем руки.

Тендосиновіт - запалення синовіальної оболонки сухожильної частини рук.

Тунельний синдром зап'ясного каналу - запалення медіального нерва руки через набряк сухожилів.

Основні заходи по запобіганню травматизму передбачені: в системі нормативно-технічної документації з безпеки праці; в організації навчання і забезпечення працюючих безпечними засобами захисту; в прогнозуванні виробничого травматизму; раціональному плануванні коштів і визначенні економічної ефективності від запланованих заходів. Основне завдання нормативно-технічної документації з безпеки праці - сприяти передбаченню небезпеки і прийняттю найбільш ефективних заходів її ліквідації або локалізації при проектуванні виробничих процесів, обладнання, будівель і споруд. Нормативно-технічна документація щодо безпеки праці розробляється з урахуванням характеру потенційно небезпечних факторів, рівня їх небезпечності і зони поширення, психофізіологічних і антропометричних особливостей людини.

Всі заходи по запобіганню виробничому травматизму можна поділити на організаційні та технічні.

Організаційні заходи, які сприяють запобіганню травматизму: якісне проведення інструктажу та навчання робітників, залучення їх до роботи за спеціальністю, здійснення постійного керівництва та нагляду за роботою; організація раціонального режиму прані і відпочинку; забезпечення робітників спецодягом, спецвзуттям, особистими засобами захисту; виконання правил експлуатації обладнання.

Технічні заходи: раціональне архітектурно-планувальне рішення при проектуванні і будівництві виробничих будівель згідно санітарних, будівельних і протипожежних норм і правил; створення безпечного технологічного і допоміжного обладнання; правильний вибір і компонування обладнання у виробничих приміщеннях відповідно до норм і правил безпеки та виробничої санітарії; проведення комплексної механізації і автоматизації виробничих процесів, створення надійних технічних засобів запобіганню аваріям, вибухам і пожежам на виробництві; розробка нових технологій, що виключають утворення шкідливих і небезпечних факторів та інше.

Важливим у забезпеченні безпечної праці і запобіганні травматизму на виробництві є фактори особистого характеру - знання керівником робіт особистості кожного працівника, його психіки і особливостей характеру, медичних показників і їх відповідності параметрам роботи, ставлення до праці, дисциплінованості, задоволеності працею, засвоєння навичок безпечних методів роботи, знання норм і правил з охорони праці і пожежної безпеки, його ставлення до інших робітників і всього колективу.

# ВИСНОВКИ

В даній роботі було проаналізовано мотиви створення системи керування метаконтентом для туроператора. Було проаналізовано актуальність проблеми. Також в роботі було розглянуто фреймворки Java Spring, а також функціональні особливості даного фреймвоврка . Було прийнято рішення використовувати фреймворк Spring в поєднанні з Hibernate для реалізації серверної частини .Було оглянуто аспектно-орієнтоване програмування ,яке частково використовується в системі.Для клієнтської частини було вирішено використовувати Bootstrap, Jquery і фреймворк Apache tiles,систему Google maps для відображення маршруту туру і систему збереження фотографії Panoramio ,за допомогою якої користувач може подивитись фотографії з місць де проходить тур . В наступній частині було оглянуто і проаналізовано існуючі системи і які особливості вони мають. Було оглянуто і проаналізовано існуючі системи керування метаконтентом дайджестового ресурсу туроператора ,їх основні функціональні можливості, а також проаналізовано основні відмінності створеної системи від інших подібних систем.

# Список використаної літератури

1. Вовк Р.Б., Шекета В.І. Методичні вказівки до дипломного проектування – Івано-Франківськ: ІФНТУНГ, 2014. – 44 с.
2. Хеник Б. HTML и CSS: путь к совершенству. – СПб.: Питер, 2011. – 336 с.: ил. – (Серия «Бестселлеры O’Reilly»).
3. Флэнаган Д. JavaScript. Подробное руководство. – Пер. с англ. – СПб: Символ-Плюс, 2008. – 992 с., ил.
4. http://uk.wikipedia.org/wiki/JQuery
5. Крейг “У. Spring in action 3” .- ДМК Пресс.,2013.-754 c.
6. http://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/
7. http://habrahabr.ru/post/222579/
8. http://docs.jboss.org/hibernate/orm/4.2/devguide/en-US/html/
9. http://en.wikipedia.org/wiki/Apache\_Maven
10. http://tiles.apache.org/framework/tutorial/index.html
11. http://getbootstrap.com/css/
12. http://en.wikipedia.org/wiki/Google\_Maps
13. http://en.wikipedia.org/wiki/Panoramio
14. http://en.wikipedia.org/wiki/Flickr
15. http://habrahabr.ru/post/165649/
16. http://uk.wikipedia.org/wiki/Аспектно-орієнтоване\_програмування
17. Стеліга І. І., Калічак О. В., Рошак Й. І., Кривенко Г. М. Охорона праці в галузі: Практикум. – Івано-Франківськ: ІФНТУНГ, 2011. – 72 с.
18. Семчук Я. М., Воєвідко І. В., Кулик М. П. Охорона праці. Начальний посібник. – Івано-Франківськ: ІФНТУНГ, 2010. – 408с.
19. Закон України "Про охорону праці" від 9.04.2014 року.
20. Протоєрейський О.С. Охорона праці в галузі: навчальний посібник/О.С.Протоєрейський, О.І.Запорожець. – К.: Книжкове видавництво НАУ, 2005. – 268 с.
21. ТретьяковО.В. Охорона праці: Навчальний посібник з тестовим комплексом на CD/О.В.Третьяков, В.В.Зацарний, В.Л.Безсонний; за ред. К.Н.Ткачука. – К.: „Знання”, 2010. – 167 с.
22. ДСанПІН 3.3.2.007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.