

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ
ФОРМ НАВЧАННЯ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

**«Інформаційна система управління відносинами з
клієнтами турагенства»**

**Завідувач
випускової кафедри**

Довбиш А. С.

Керівник роботи

Шелехов І.В.

Студент гр. ІІз-61С

Баштовий К.

Суми 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ НАВЧАННЯ
Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2020 р.

ЗАВДАННЯ

до випускної роботи

Студента четвертого курсу, групи ІНз-61С Баштового Костянтина спеціальності “Комп'ютерні науки” заочної форми навчання.

Тема: “Інформаційна система управління відносинами з клієнтами турагенства ”

Затверджена наказом по СумДУ

№ _____ от _____ 2020р.

Зміст пояснювальної записки: 1) характеристика та аналіз предметної галузі; 2) вибір методу проектування архітектури та моделі функціонування системи; 3) проектування компонентів системи

Дата видачі завдання “ _____ ” _____ 2020р.

Керівник випускної роботи _____ Шелехов І.В.

Завдання прийняв до виконання _____ Баштовий К.

.

РЕФЕРАТ

Записка: 72 стор., 27 рис., 3 табл., 1 додаток, 10 джерел.

Об'єкт дослідження – процес проектування інформаційної системи управління відносинами з клієнтами.

Мета роботи — розробка інформаційного та програмного забезпечення інформаційної системи управління відносинами з клієнтами.

Методи дослідження — технологія проектування інформаційних систем, методи моделювання бізнес-процесів, технологія проектування баз даних, технологія проектування веб-орієнтованих додатків.

Результати — розроблено та програмно реалізовано інформаційної системи управління відносинами з клієнтами турагенства. При цьому було застосовано такі мови програмування, засоби розробки та фреймворки: HTML5, JavaScript, JQuery, PHP5 (Laravel Framework), MySQL та CSS3. Розроблена багатофункціону систему управління взаємодії з клієнтами дає змогу автоматизувати ведення справ у туристичному агентстві та підвищує продуктивність підприємства та його привабливість для клієнтів

СИСТЕМА CRM, СИСТЕМА КЕРУВАННЯ БАЗОЮ ДАНИХ,
HTML5, JAVASCRIPT, JQUERY, PHP5, LARAVEL FRAMEWORK,
MYSQL, CSS3

ЗМІСТ

ВСТУП	5
1 ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	7
1.1. Характеристика предметної галузі та об'єкта дослідження	7
1.2. Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі.....	11
2 ВИБІР МЕТОДУ ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА МОДЕЛІ ФУНКЦІОНУВАННЯ СИСТЕМИ.....	20
2.1 Аналіз системних вимог та обґрунтування методу проектування системи	20
2.2. Постановка та алгоритм розв'язання задачі.....	21
2.3. Архітектура запропонованої системи	24
3 ПРОЕКТУВАННЯ КОМПОНЕНТІВ СИСТЕМИ	27
3.1. Інформаційне забезпечення систем.....	27
3.2. Технічне забезпечення.....	46
3.3. Програмне забезпечення	48
3.4. Організаційне забезпечення.....	60
ВИСНОВОК.....	63
СПИСОК ЛІТЕРАТУРИ.....	64
ДОДАТКИ.....	65

ВСТУП

Ринок туристичних перевезень в Україні кожен рік зростає і є популярним напрямком діяльності. Ця ніша стрімко розвивається і на українському ринку є багато зарекомендованих компаній, що займаються туристичними перевезеннями. За рахунок підвищення кількості перевезень туристичні фірми потребують системи взаємовідносин з клієнтами для покращення бізнес-процесів й зростання якості надання послуг.

В наші дні збільшується конкуренція серед туристичних підприємств. Кожне з них розвивається і намагається запропонувати все більше різних послуг клієнтам в умовах невизначеності ринку. Для задоволення більшого попиту без зниження якості обслуговування клієнтів потрібна єдина система, що дозволяє контролювати роботу туристичної фірми так, щоб задовольнити потреби, індивідуальні смаки кожного клієнта і спростити процес співпраці з ними. CRM система туристичної фірми може зберігати і обробляти велику кількість даних про покупців, яку потім можна використовувати для поліпшення якості сервісу і поліпшення взаємин з клієнтами. [1]

Керування взаємовідносинами зі споживачами ґрунтується на принципах маркетингу відносин, що розвивається в області сучасного маркетингу.

Це було доведено до відома керівників туристичних фірм з початку 2000-х років. Всі результати були засновані на перевагах і вимогах їх клієнтів. Ця інформація була використана для підвищення задоволеності клієнтів наданими послугами і утримання їх для подальшого співробітництва. У тому ж десятилітті висувалося припущення, що в майбутньому фірми не будуть намагатися продати один і той же товар чи послугу багатьом клієнтам, а вони будуть намагатися продати більшу кількість своїх продуктів одному клієнту за тривалий період часу. Такий підхід проклав шлях для переходу від RM до CRM. Підхід CRM концентрується на створенні унікальних відносин з окремими клієнтами на основі управління взаємовідносинами з певним клієнтом конкретно для того, щоб забезпечувати і керувати, а головне створювати довгострокові відносини.

На даний момент, туристичні фірми стикаються з ще більшою конкуренцією на ринку, і існує потреба в тому, щоб запропонувати найбільш широкий спектр послуг для різноманітної клієнтської бази. Тому, впровадження CRM може служити вкрай необхідним стратегічним рішенням, яке дозволить власникам турфірм налаштувати процес своєї роботи таким чином, щоб індивідуальні забаганки клієнтів були задоволені, а також полегшився процес обробки замовлень. Було доведено, що туристичні підприємства, які впровадили CRM отримують велику вигоду в результаті пошуку і утримання постійних клієнтів, адже вартість повторного залучення до покупки існуючого клієнта нижче, ніж залучення нового. Робота на утримання клієнта з компанією дозволяє збільшити лояльність споживача для самої компанії.[2]

1 ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Характеристика предметної галузі та об'єкта дослідження

Усі програмні й інформаційні системи, що застосовуються в ході автоматизації відповідного процесу взаємостосунків із замовниками, слід вважати винаходом тих осіб, для кого одна з мотивуючих цілей звучала таким чином: «Як автоматизувати нераціональний, слабо організований та дорогий процес збуту?».

З тих часів подібні ранні прототипи даних програмних продуктів могли допомогти певному продавцю у його оперативній роботі. Наприклад, такими торговцями були особи, як Джон Генрі Паттерсон, що створив фірму National Cash Register. Він був першим хто разом поєднав способи, які використовуються для системи торгівлі. Поряд за ним в 1911 р. вийшов Джем Ханді, що робив на Chicago Tribune. Він досліджував причини, що чинять вплив на покупку клієнтами тих або інших продуктів. Він стверджував, саме, що застосування відповідних банерів та рекламних роликів, що пояснюють споживачам позитивні сторони товарів та послуг, які просуваються, надасть виробникам перевагу щодо тих, хто діє відповідно до звичайної на той час системи торгівлі. Далі їм була сформована фірма «A Jam Handy Production», та вона застосовувалася більш ніж в 7000 типах рекламних товарів.

У 1947 р. Адвокат Морріс Перкін створив систему запису планів бізнесових зустрічей, а саме Day-Timer. Вона вміщувала засоби масової інформації для майбутніх подій. Система функціонувала як щоденник, де можливо запланувати час, а також де було відмічено роботу.

Цю схему вже можна було б віднести до прототипів сучасних систем CRM. З 1952 року Деймер-Таймер набув значного використання. У 1960 році розвиток системи досяг Великобританії та Канади.

Ще ближчим до комп'ютерних технологій сьогодні є Арнольд Нустадтер, науковець Relodex. Це був 360-градусний обертовий файловий кабінет, який до 70-х років був найбільш використовуваним інструментом для запису бізнес-контактів, зустрічей тощо. Поряд з такою подією розпочалася ера комп'ютерів.

але оскільки машинний час залишався дорогим, більшості компаній довелося обмежитися такими інструментами, як Day-Timer і Roldex.

На початку 80-х кількість систем СРМ була надзвичайно малою, але зростання комп'ютерних технологій дозволив знизити витрати на обчислення. У 1987 році Пет Салліван виявився засновником Solar Software International і SolarLogis Software і запустив піонерську комерційну комп'ютерну програму керування контактами, відому як «АСТ!». Ця програма дозволила продавцям стежити за розвитком відносин із клієнтами, обробляти та оцінювати інформацію про ці відносини. Поняття "CRM-система" набуло абсолютно нову якість до середини 1990-х. Система CRM розумілась, як автоматизація технологій продажу, орієнтованих на клієнтів. Саме поняття CRM вперше було використане Sybel Systems для відображення специфіки цього типу корпоративного програмного забезпечення.[1]

В окремому ланцюгу відносин із клієнтом такі системи займалися в період від найпершого контакту з клієнтом до кінцевого ухвалення угоди. Давайте докладніше розглянемо їх[2]:

- керування різними маркетинговими системами, висвітлення повної інформації щодо події, контактної аудиторії, вимірювання всіх витрат, оцінку результативності;
- ведення бази клієнтів, розпочинаючи з прояву першого інтересу до компанії, до постійних клієнтів.
- документування певного процесу обслуговування клієнтів.
- засоби аналізу перспектив роботи з покупцями.

Зі створення мереж такі інформаційні системи збільшились та отримали статус систем автоматизації роботи відповідних підрозділів компанії. Наступний етап розвитку в даному напрямку стала поява «Повних» CRM-систем, що можуть автоматизувати всю роботу організації в широкому діапазоні взаємовідносин із клієнтами.

Поняття CRM визначається як бізнес-стратегія компанії, спрямована на клієнтів і що вимагає узгоджених дій з боку всіх працівників компанії, а також

процесів та технологій побудови й розвитку взаємовідносин з покупцями з ціллю зростання прибутку. Основою для концепції CRM є індивідуальний підхід до кожного клієнта компанії, і також акумулювання усієї доступної інформації та даних про них, для надання товарам найбільшій споживчій цінності та забезпечення рівня попереднього продажу та після продажного сервісу, бажаного покупцем.

Іншими словами, концепція CRM не спрямована безпосередньо на збільшення продажів, а на забезпечення досягнення цього позитивного ефекту шляхом покращення та зміцнення відносин із клієнтами для підвищення їх задоволеності продуктами та послугами.

Роль інформаційних технологій у впровадженні системи CRM полягає в ефективному використанні всіх каналів зв'язку з клієнтами та замовниками для акумулювання, зберігання, обробки та оцінки різних отриманих даних.

На основі зібраних даних, таких як, наприклад, історія покупок або уподобання покупців, прогнозується поведінка клієнтів, фактори, які керуватимуть компанію збільшувати утримання та формування клієнтів та їх ранжирування з метою визначення пріоритетності найбільш прибуткових запитів клієнтів та кількості важливих завдань.

Система CRM, що представляє собою набір програмних модулів, та повинна містити такі інструменти[3]:

- автоматизація територіально розподілених продажів.
- замовлення та виставлення рахунків в Інтернеті.
- ведення електронних каталогів товарів.
- аналітична обробка інформації щодо покупців.
- проведення маркетингових досліджень.

Є два основні напрями функціонування CRM-систем: автоматизація збуту та обслуговування споживачів. До функцій першого напрямку належать:

- керування контактами: підтримка інформації про покупців та історії взаємовідносин, враховуючи інформацію про частоту, види та обсяги придбаних товарів.

- керування діяльністю: графік бізнес-активності регіональних торговельних представників.
- керування зв'язком: існування надійних каналів зв'язку.
- прогнозування: підготовка плану прогнозу збуту на базі власних даних та інформації маркетингових досліджень профільних компаній.
- керування можливостями: управління чинниками залучення конкретних клієнтів.
- керування замовленнями: отримання потрібної інформації щодо наявності товарів на складі та організація їх доставки чи переробки виробничого замовлення вимагає продукції.
- керування документацією: створення та впровадження потрібних форм звітності, інформації та рекламних даних.
- оцінка продажів: збір, показ та аналітичне оброблення даних щодо збуту.
- зберігання даних про різні товари та їх ціни.
- енциклопедія маркетингу: наявність актуальної інформації про товар, ціноутворення, акції, результати маркетингових досліджень, інформація про конкурентів та їхню продукцію.

Обслуговування клієнтів включає[3]:

- управління обробкою запитів покупців, автоматична обробка запитів, пошук відгуків покупців про рівень сервісу, якість продукції та нові вимоги до неї.
- управління сервісом: запис даних, затрат та часу, пов'язаних із обслуговуванням клієнтів, надання постачальникам перевічених рішень зі створеної та оновленої бази знань.
- швидке обслуговування інформації про покупців: представлення інформації щодо нових видів товарів, відповіді на запитання, реєстрація й оброблення виникаючих проблем. Очевидно, що одним з найбільш важливих чинників успіху впровадження систем CRM є здатність швидко, майже негайно організувати, можливість відповідати на зміни вимог клієнтів та виконувати запити у відповідності до новітніх критеріїв договорів. Для вдалого виконання

таких завдань виробничою діяльністю підприємства необхідно керувати і чітко планувати. Поряд з цим з цим пропонується формування системи CRM здійснювати далі або одночасно із впровадженням систем MRP або ERP.

CRM-система - комерційна інформаційна система, яка служить для покращення обслуговування покупців на основі використання методу зберігання інформації про покупців та історії їх взаємозв'язків, виявлення та покращення потоку бізнес-процесів шляхом зберігання даних та подальшої діагностики їх ефективності[3-4].

Поняття CRM можливо перекласти на українську мову як "управління відносинами з клієнтами". Цей термін має на увазі[4]:

- наявність загального сховища інформації, де в любий час можливо отримати всю інформацію щодо взаємодії з покупцем.
- синхронізація управління декількома каналами зв'язку (Це передбачає наявність організаційних процедур, що встановлюють правила застосування такої системи й інформації у відповідному відділі компанії).
- систематичний аналіз представленої інформації про відносини з клієнтом й ухвалення певних організаційних рішень.

Цей підхід до справи має на увазі, що при любій взаємодії зі споживачем працівник будь-якого каналу компанії має доступ до релевантної інформації про всі стосунки з покупцем та рішення щодо подальшої роботи ухвалюється на його основі.

1.2. Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

Оскільки система CRM є веб-додатком, вона потребує веб-сервера та системи керування базами даних (СУБД) [5-7]. На робочому місці клієнта не потрібно встановлювати наявне програмне забезпечення, крім стандартного браузера. Ідеологічна основа для роботи системи CRM має бути представлена у формі двох баз даних, які обслуговують усі етапи взаємовідносин із клієнтом. Перша база даних - MySQL, яка містить дані про всі транзакції, які не

потребують роботи в Інтернеті, а друга база даних - це сервер для роботи в Інтернеті з багатьма клієнтами. Це допомагає бронювати місця та приймати замовлення. Реалізація взаємодії клієнт-сервер означає, що як для зовнішньої, так і для внутрішньої, відносно компанії, користувачі мають абсолютний доступ до даних, що зберігаються в архівованій базі даних, через веб-браузер. В аналітичній частині функціональності CRM-сервера є два компоненти: СУБД для зберігання даних, а також основна обробка інформації та засоби OLAP - розробка способів обробки даних, що містить формування та динамічну публікацію звітів та документів. використовується фахівцями для оброблення складних запитів до бази даних якнайшвидше та для аналізу інформації в Інтернеті. Найбільше використання OLAP знаходиться в результатах бізнес-планування й сховищах інформації. Система CRM реалізована як веб-додаток, що дає можливість працювати з нею на різних платформах, в тому числі і на мобільних пристроях. Робоче місце здатне бути як на одному пристрої, так і на декількох. Система CRM не має ресурсної залежності, тому потужність звичайного комп'ютера чи ноутбука достатня для раціональної роботи інформаційної системи. Багато CRM-системи народилися із систем, що автоматизували відповідні процеси взаємовідносин з клієнтами. Багато існуючих CRM є нащадками систем SFA, SMS та CCS. Раніше системи CRM використовувались як системи для автоматизації відповідних служб тощо. Проте у середині 2000-х років система CRM отримала всі ці можливості обробки інформації. Видів CRM-систем надзвичайно багато. Найпопулярніша на сьогодні класифікація - це поділ відповідних CRM на посилення на обробку інформації та завдання, які вирішуються організаціями в ході взаємодії з покупцями. Операційний CRM - це рівень, де реалізується автоматизація операційних процедур виконавців та керівників. реєстрація та швидкий доступ до певної інформації щодо подій, процесів, проектів, контактів тощо. В Україні, власне, інформаційні системи такого рівня отримали назву CRM-систем. До даного рівня слід віднести фронтіві системи маркетингу, автоматизації збуту та сервісу. Системи інтеграції фронтівих та бухгалтерських підсистем; насправді,

бухгалтерські системи, які зберігають та обробляють фінансову інформацію щодо клієнтів. Достатньо подивитися список основних особливостей CRM даного класу: Збір, зберігання, класифікація різної інформації про покупця. Формалізація та підтримка усіх головних бізнес-процесів взаємовідносин з клієнтом. Це здатні бути контрактні процеси, реєстрація замовлень клієнтів, збут, групове сповіщення покупців через різні канали зв'язку тощо. Те саме стосується торгового центру. Істотною частиною підтримки цих процесів є дані про фазу процесу (проміжна точка). Наприклад, переговори, клієнт одержав договір на ухвалення тощо. Функції для планування та контролю завдань обслуговування клієнтів. Розвиток маркетингових функцій (опитування клієнтів та опитування, збір даних щодо цін конкурентів тощо, аналіз рекламної компанії). Цей список можна продовжувати, але головна ідея зрозуміла: цей клас систем розроблений для щоденного обслуговування клієнтів, організованого максимально ефективно. Зазвичай такі системи орієнтовані на детальний збір даних, але не на аналіз. Споживачами наслідків автоматизації даного рівня вважаються прямі підрядники - працівники служб, які працюють із споживачами: маркетологи, менеджери зі збуту, секретарі, менеджери відповідних лінійних відділів, менеджер з продажу, відділ підтримки клієнтів. Важливо відмітити те, що такі системи забезпечують міцну інтеграцію між відповідними підсистемами. Аналітичним CRM вважається рівень автоматизації процесів управління, пов'язаний з глибокою оцінкою клієнтської бази клієнтів. Інформаційні системи даного рівня - це системи зберігання даних, а також оперативний та статистичний аналіз бази даних покупців. Ці системи забезпечують звітність за первинними даними та ретельний аналіз отриманої інформації. Споживачами продуктів аналітичної автоматизації рівня CRM є управлінці, що відповідають за стратегічне управління компанією. Ми виділяємо основні критерії, за якими систему CRM можна класифікувати як аналітичну: Першочергово, це оцінка продажів у різних розділах. Аналіз товарного асортименту (наприклад, на базі даних про збуту). Аналіз системи ціноутворення, включаючи оцінку конкурентного середовища. Тісна інтеграція

з системою обліку буде складовою частиною. Оцінка товарних залишків (потреба в постачанні, неліквідні активи, строки зберігання тощо). Особливості прогнозування збуту. Класифікація й сегментація покупців (виявлення найцікавішого, своєчасного запобігання відходу клієнтів до конкурентів тощо). Перелік задач, що вирішуються на основі цього CRM, включає синхронізацію даних та пошук їх статистичних зразків для створення найбільш раціональної маркетингової стратегії, збуту та сервісу клієнтів. Для аналізу потрібна велика кількість статистики, ефективний інструмент для аналізу даних та хороша інтеграція системи. Але варто зазначити, що аналітичний CRM не настільки широко розповсюджений, як оперативний. Її основи відповідають поняттям DataWarehousing, Data mining, тому постачальники систем у таких сферах активно просувають і репозиціонують свої системи як аналітичні CRM системи (наприклад, SAS). Дані, згенеровані такими системами, успішно використовуються відділами маркетингу, а також здатні бути представлені покупцю. Проте ніхто не виключає поєднання двох описаних вище систем. Нова система одержала назву CRM для керівництва фірмами. вони дозволяють визначати та орієнтуватися на довгострокові групи клієнтів на основі рекламних повідомлень. Зазвичай використовуються такі канали: традиційні та електронна пошта, телефон, SMS. Системи наділені функціями зберігання й обробки одержаних даних після проведення подібних дій. Потім формується статистична база даних. Спільна CRM (Співпраця; Спільна, узгоджена дія) - цей рівень вважається нішею, де здійснюється автоматизація тісних стосунків з клієнтами з різних каналів зв'язку (електронні засоби, телефон, прямий контакт). Такий перелік інформаційних систем здатний скласти веб-портал, електронну пошту, торговий центр, систему обліку та планування контактів. Такі системи встановлюють тісніший контакт з клієнтом, що здатні впливати на окремі процеси безпосередньо всередині самої організації. Це створює умови для постійного доступу клієнта до внутрішніх процесів організації. вони можуть включати опитування щодо зміни якості продукту або послуги, створення веб-сторінок для відстеження руху клієнтів замовлень, SMS-

повідомлень про завершені операції з банківськими рахунками, можливість споживачів формувати та замовляти себе в режимі реального часу, наприклад, доступні системи автомобілів чи обладнання та варіанти та багато іншого. Споживачами результату цього рівня автоматизації є ті ж працівники, що і в операційній системі управління CRM. Системи, що підтримували б спільні CRM, фактично відсутні на ринку одночасно, оскільки процес спільної роботи у багатьох випадках індивідуальний і повинен автоматизуватися надзвичайно гнучкою системою CRM [5].

Крім цього, така інформаційна система має містити, по суті, найдешевшу та найвигіднішу технологію для вкорочення витрат на розробку інтерфейсу між фірмами та замовниками. Управління продажами. При плідної взаємовідносин з будь-яким клієнтом фірма може формувати для них найбільш сприятливі умови, тим самим збільшуючи їх прибуток. Раціональне управління готівкою неможливе без використання CRM, тому-що ці системи дають можливість точно спланувати ймовірність укладення контрактів, а отже, збільшувати їх. Зниження витрат досягається на основі позбавлення від звичайних процедур, на які працівники витрачають чималий час. Плинність кадрів зменшується персоналом. Кожен працівник може порівняти результат власної праці з роботою співробітника. Це дозволяє перейняти гарний досвід клієнта. Управління маркетингом. На основі системи планування та оцінки компанія має можливість проводити прямий маркетинг. Рекламні кампанії базуються на вибірці клієнтів. Управління сервісом та торгові центри (системи подання скарг клієнтів, отримання та подальше обслуговування комунікацій з клієнтами). Системні операції автоматизовані, що покращує процес контролю замовлень, зменшуючи незаплановані витрати. На основі системи можна покращити якість обслуговування, а також підвищити лояльність клієнтів. Після контакту та отримання якісної послуги існуючий замовник повернеться до продовження співпраці з компанією. Однак найпопулярніші CRM системи, що формують таку функціональність. В останні роки модель продажу CRM за запитом набула широкого поширення у світі. Найчастіше переваги від втілення CRM

сприймаються людьми на керівних посадах на рівні здорового глузду. Вочевидь, таких позитивних факторів, як утримання клієнтів, фахівців навіть не розуміє. Труднощі при оцінці ефективності впровадження системи CRM виникають при спробі точно визначити взаємозв'язок між інвестиціями та рентабельністю (Return on Investments, ROI), оскільки не існує єдиної формули, що могла б слугувати в цій оцінці. Нині вже не секрет, що ефективність інформаційної системи відображається її наповненням й якістю впровадження. Іншими словами, від якості сформованої бізнес-моделі залежить правильність структур та організації процесів, або, кажучи іншим чином, результат втілення рішення CRM. Водночас, на практиці розробка CRM в більшості випадків втілюється як автоматизація існуючих процесів з наявним персоналом. Якщо якість бізнес-моделі за цих умов підходить, то цей підхід цілком усвідомлюється. У цьому випадку ми можемо почати говорити про ефекти впровадження: підвищення продуктивності працівників, зростання швидкості сервісу, усунення втрат та дублювання наявної інформації. Він оцінює вплив придбання CRM-програми та його впровадження з урахуванням поточних процесів в організації. Також, слід сказати, що із впровадженням системи CRM як програмного забезпечення для автоматизації процесів фірми одержувати прямі ефекти скорочення витрат та деякі непрямі ефекти, отримані завдяки підтримці існуючої бізнес-моделі. Прямі економічні наслідки. Ця категорія зазвичай включає ефекти прямої дії, від яких залежить результат фірми. Наприклад, сегментація покупців, вибір каналу та оптимальна ланцюжок просування, раціоналізація організаційної структури, формування системи мотивації працівників тощо. В результаті втілення CRM з'являться короткотермінові та довгострокові ефекти, зокрема, збільшення продажів шляхом фокусування щодо прибуткових клієнтів, а також збільшення доходу організації через більш вигідні сегменти. [6]

Непрямі економічні наслідки. До цієї категорії належать корпоративні ефекти, які важко визначити шляхом прямого розрахунку і тому мають особливе значення для акціонерів компанії. Наприклад, ці ефекти можна

віднести до збільшення вартості акцій на біржі за рахунок підвищення ступеня прозорості процесів, зростання ступеня керованості. Це важливо для залучення нових акціонерів. Для вимірювання ефекту від втілення CRM можна застосувати метод аналізу декількох ключових характеристик до та після (і в ході) змін. Це вимірювання, за допомогою яких фірма надалі оцінить ефективність власних стосунків із клієнтами. Деякі з цих характеристик можна віднести до більшості фірм до початку плану. Вибрано ряд особливостей компанії, наприклад: відсоток резонансу потенційних покупців до реклами (відповідь аудиторії); зростання нових клієнтів (норма прибутку); ціна покупки; жменька успішних угод; тривалість циклу збуту; середній час для виконання типових завдань на обслуговування покупців тощо. Характеристики, як правило, групуються за групами бізнес-процесів чи підсистемами CRM. Явище формалізованої оцінки ефективності втілення CRM полягає в тому, що нефінансові дані потрібні за період до впровадження системи CRM, і ці дані можуть лише допомогти системі CRM зібрати. Так, можна просто оцінити сухий баланс - підвищення прибутку компанії в різні періоди часу. Проте це спричинено впровадженням CRM-рішення. Щоб відповісти на це запитання, вам потрібно вміти розуміти структуру клієнтської бази, ефективність роботи менеджерів, підвищення лояльності клієнтської бази та багато іншого, насправді, що можна створити за підтримки системи CRM себе. Як результат, для обґрунтованої оцінки вибрані характеристики (як природні, так і вартісні) вже відстежуються під час реорганізації процесів та втілення компонентів інформаційної системи. Існує також можливість порівняння вираження наслідків реорганізації грошових коштів та відповідних витрат для оцінки періоду окупності інвестицій в інформаційну систему управління ресурсами. Водночас, інша проблема в оцінці ефективності виявляється в тому, що для будь-якої конкретної фірми впливають однакові фінансові результати від втілення рішення щодо управління ресурсами. Не маючи інструментів на місці, майже кожен дає оцінку високої якості. Наприклад, "відсоток утримання клієнтів збільшився на 5-10%, фактично це призвело до збільшення 20-30%,

автоматизації маси ручних операцій майже вдвічі збільшило продуктивність персоналу" та ін. Усі ці практичні оцінки мають цінність. Отже, яка можливість оцінити ефекти від впровадження до запуску проекту? Є така можливість. Все можна досягти, реалізуючи певну бізнес-модель. Фактично цю модель потрібно розробити на ранньому етапі підготовки впровадження системи CRM. Далі така модель працюватиме як орієнтир, що підтверджує досягнення характеристик, які вбудовані в неї. Складність впровадження нових рішень CRM лякає багатьох фірм, що зацікавлені в їх застосуванні, що не дивно, адже будь-який керівник бізнесу хоче отримати вигоду лише від інновацій. Звичайно, до моменту появи систем CRM вже існують загальноприйняті методи прийняття, які можна використовувати для досягнення успіху при впровадженні CRM, проте якщо використовуватись без певних умов, ви можете тільки принести більше негативних наслідків, ніж користі. Якщо, однак, процес реалізації технології CRM ефективно розвивається і виникає питання щодо придбання засобів автоматизації, є окремі загальні критерії, що слід враховувати при виборі таких інструментів. Основні з них включають: Термін реалізації. Складність. Гнучкість. Безпека. Архітектура та багатство клієнтської бази. Масштабованість. важливість інтеграції, яких існує два види: 1) бізнес: між каналами, підрозділами, функціями тощо. 2) технічні: стандарти, підтримка інтеграції баз даних, функціонування Інтернет-інструментів. Підтримка комунікацій. Прості у використанні та функціональні. Якщо система стане незручною у використанні, то проект буде неефективним. Тому потрібно створити систему зручною, пояснити переваги інновацій та забезпечити навчання кадрів. Кожна організація має власні очікування та ідеї, і, відповідно, вимоги до створюваної CRM-системи. Ось чому не може бути рішення одного розміру. Питання має вирішуватися індивідуально. Іншою складністю після вибору системи для фірми буде проблема впровадження. Тут є проблеми [6-7].

Ось окремі з них: Користувачі (мала фаховість продавців, техніків та диспетчерів). Неформальні бізнес-процеси. Швидкість змін в керуванні відносинами з клієнтами. Непередбачений вплив зовнішніх факторів

(політики). Розвиток засобів зв'язку. Недостатнє фінансування. Створити повноцінну команду, яка складається із залучених до процесу впровадження власних менеджерів та спеціалістів та зовнішніх консультантів. Оцінка вартості та часу проекту двічі є найбільш оптимістичною та найбільш песимістичною. Використовуйте теорію ймовірностей для оцінок. Оцініть проект на основі аналогій. Оцініть проект по максимуму. Корируйте кошториси проектів у міру їх виконання. Розділіть проект на окремі компоненти та реалізуйте його по частинах. Перевірте всі базові технічні рішення у нових проектах. Якщо кожна організація, у відповідності до концепції CRM-систем, має бути цілком відкритою для своїх клієнтів та клієнтів, то до кого вона повертається в іншу сторону? Ми можемо знайти лише одну відповідь - до Постачальника. Усі компанії добре розуміють, що вони своїм покупцям можуть запропонувати тільки те, що вони купують у постачальника. Тому пілотні системи Surly Chain (ланцюги постачання) розробляються паралельно із системами CRM майже з однаковою швидкістю. Якщо поєднати ці два новітніх рішення, ви отримаєте якусь досконалу систему, яка автоматизує всі відносини компанії. Такі системи називаються ERMCRM + SurriereRM = Enterrière RM. Ще один шлях розвитку, який пройшли практично всі виробники ERP, - це розгалуження таких систем CRM з функціональністю. ERP + CRM = IRP (планування ресурсів розвідки). Можна більше стверджувати про можливості інтеграції CRM-систем, про новітні напрями розробки стратегій, але головне не забувати, про що йдеться, не випадати разом із теорією замовника. Можна забути формули, проте слід пам'ятати, що добробут компанії буде чіплятися за своїх Клієнтів, і це завжди потрібно.

2 ВИБІР МЕТОДУ ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА МОДЕЛІ ФУНКЦІОНУВАННЯ СИСТЕМИ

2.1 Аналіз системних вимог та обґрунтування методу проектування системи

Щоб спроектувати інформаційну систему потрібно визначити та проаналізувати вимоги до системи, що зможуть значно полегшити процес проектування та внести необхідні уточнення. Вони являють собою по суті базовий набір правил та обмежень, яким має відповідати ІС, для виконання покладених на неї функцій. Зазвичай такий набір принципів встановлюється в технічному завданні, яке підлягає попередньому обговоренню двома сторонами: замовником та виконавцем. Згодом після затвердження ТЗ, ним скористаються вже розробники ІС при побудові системної архітектури.

Для задачі аналізу резюме кандидата можна встановити такі вимоги:

1. Система має бути зручною для користування в таких аспектах:
 - 1.1. Введення інформації про систему клієнтом.
 - 1.2. Виведення інформації про надходження на екран.
 - 1.3. Трьохрівневе делегування доступу.
2. Система має оперувати тільки інформацією, яка пройшла валідацію під час вводу.
3. Система має встановити значення по замовчуванням, або якщо були введені невірні дані то вказати на це та не допустити введення інформації до моменту виправлення.
4. Система має надати зручний доступ до послуг зі сторони клієнтів.
5. Система має надати змогу переглядати інформацію за кожним клієнтом.
6. Система має мати функціонал інформування клієнтів через електронну пошту.

Дана ІС спроектована як система керування зв'язку з основними клієнтами, що містить підсистеми інформації.

Інформаційні підсистеми є основною складовою частиною комплексу інформаційного забезпечення й структури інформаційної системи керування

призначені для збору, накопичення, обробки й зберігання інформації відповідних напрямів обліку та застосовуються в роботі численних служб.

Інформаційно-аналітична система (ІАС) вважається комп'ютерною системою, що дає можливість одержувати інформацію, формувати її й здійснювати власне обробку й аналіз.

Система підтримки ухвалення рішень – автоматизована система, що шляхом збору й аналізу значної кількості інформації здатна впливати на процес ухвалення управлінських рішень в бізнесі та підприємстві.

Таблиця 2.1 – Етапи й стадії формування ІС управління зв'язку з клієнтами

Стадії	Етапи робіт
1. Розробка вимог до ІС	1.1. Виявлення об'єкта оцінки й обґрунтування потреби створення ІС. 1.2. Визначення вимог користувачів до ІС.
2. Формування концепції ІС.	2.1. Дослідження сфери туристичних послуг. 2.2. Здійснення потрібних науково-дослідних робіт. 2.3. Формування варіантів концепції ІС, які задовольняють потребам користувачів. 2.4. Створення звіту щодо виконаної роботи.
3. Технічне завдання.	3.1. Розробка та затвердження технічного завдання на формування ІС з метою автоматизації туристичного підприємства
4. Ескізний проект.	4.1. Визначення перших варіантів системи управління на базі існуючого досвіду. 4.2. Розробка документації на ІС та відповідні модулі.
5. Технічний проект.	5.1. Формування проектних рішень за системою та її модулях. 5.2. Створення документації на ІС та відповідні модулі.
6. Робоча документація.	6.1. Розробка робочої документації щодо системи та модулів. 6.2. Створення додатку.
7. Втілення в життя.	7.1. Завантаження ІС на веб сервер. 7.2. Ознайомлення персоналу з ІС. 7.3. Здійснення попередніх випробувань. 7.4. Здійснення дослідної експлуатації. 7.5. Здійснення приймальних випробувань.
8. Супроводження ІС	8.1. Проведення робіт у відповідності з гарантійними зобов'язаннями. 8.2. Післягарантійне обслуговування.

2.2. Постановка та алгоритм розв'язання задачі

Задача «Проектування системи управління взаємодії з клієнтами туристичного агентства» призначена для спрощення керування компанією за

допомогою автоматизації та каталогування інформації про надходжуючу інформацію.

Техніко-економічний зміст задачі полягає в тому, що на базі даних, які надходять до системи від клієнтів, проводиться розрахунок то доцільність певної подорожі .

Перелік об'єктів, при роботі з якими працює система: клієнти туристичного агентства.

Вихідна інформація використовується відділом роботи з клієнтами туристичної компанії, відділом зв'язків з компаніями-партнерами та бухгалтерією підприємства.

Періодичність роботи системи – відповідно до періодичності надходження заявок від клієнтів.

Повний цикл роботи системи завершується, коли усі етапи підбору потрібних послуг клієнту на основі його вхідних даних буде узгоджено з компаніями партнерами та клієнт відповідь згодою.

Модель функціонування системи представлено на рис. 2.1.



Рисунок 2.1 – Інформаційна модель системи управління зв'язку з клієнтами

Зберігання, оброблення інформації й видачу результатів проводить веб сервер, обробку результатів виконує менеджер компанії та компанії-партнери.

Вихідні дані

Перелік і опис вихідних повідомлень наведено в таблиці 2.2.

Таблиця 2.2 – Перелік та опис вихідних даних

№ з/п	Назва вих. повідомлення	Ідентифікатор	Форма подавання і вимоги до неї	Періодичність	Термін видання, час затримки	Користувачі
1.	Інформація про надходженню нового замовлення	-	Вивід на екран	Після завершення ітерації замовлення зі сторони клієнта	Будь-який	Менеджер з замовлень, компанії-партнери
2.	Інформація про відмову клієнта від замовлення	-	Вивід на екран	Після відміни бронювання клієнтом через сгенерований URL у листі	Будь-який	Менеджер з замовлень, компанії-партнери
3.	Інформація про успішне складання договору за послугою	-	Вивід на екран	Після узгодження вимог усіх сторін	Будь-який	Менеджер з замовлень, бухгалтерія

Структурними одиницями інформації, що являються питомними для функціонування системи системи є:

- тип туристичної поїздки, яку бажає клієнт;
- бажані країни чи кліматичні зони;
- бюджет, який може дозволити собі клієнт;
- наскільки підходить кейс клієнта до можливостей, що надають компанії-партнери на бажаний діапазон дат;
- укладання договору з компаніями-партнерами після надходження коштів;
- укладання договору з клієнтом та подальше супроводження.

Вхідні дані

Перелік та опис вхідних даних наведено в таблиці 2.3.

Таблиця 2.3 – Перелік та опис вхідних даних

№ з/п	Назва	Ідентифікатор	Форма подання	Термін і частота надходжень	Джерело
1.	Введення інформації про бажані умови поїздки клієнтом	CLIENT	Вхідні дані	Залежить від якості реклами	Клієнт
2.	Передача і перевірка інформації наявності у компаній-партнерів	MEETING	Вхідні дані	Після заповнення заявки клієнтом	Менеджер з замовлень

Результати роботи системи

Результатів роботи системи - три.

Перший результат - це успішний, коли інформація надана клієнтом після її перевірки на можливість задоволення у компаній партнерів підтверджується и сторони виходять на рівень підписання договорів та подальшого виконання послуги.

Другий результат - не успішний, коли інформація надана клієнтом після її перевірки на можливість задоволення у компаній партнерів не може бути надана у вигляді послуги й клієнт не хоче вносити корективи у свої плани.

Третій результат - пат, коли інформація клієнта після її перевірки у компаній-партнерів на можливість задоволення не може бути надана у повному обсязі, але сторона клієнта згодна піти на поступки, або передбачала деякі зміни зі своєї сторони, та сторони можуть вийти до консенсусу у рішенні, що призводить до першого результату.

2.3. Архітектура запропонованої системи

Архітектура CRM туристичного підприємства відображається як система взаємопов'язаних елементів цієї системи та зв'язків між ними. Архітектура CRM включає певні елементи: підсистеми, комплекси завдань, відповідні задачі як виділені технологічні процеси, технічні операції .

Визначимо, за якими критеріями, будуть виділятися структурні елементи [7]:

- основні компоненти забезпечення роботи системи;
- функції, що виконує система та модулі;
- залежність між надходженням задач до системи та реагуванням на них;
- інтеграція CRM з іншим ПЗ компанії;
- зв'язок з API компаній-партнерів для автоматизації процесів;
- делегування доступу до модулів системи;
- використання процесів, що відбуваються у системі керування, у циклі роботи компанії;

Враховуючи ці критерії можна виділити такі модулі системи:

1. Модуль прийняття клієнтських заявок зі сторони сайту.
2. Модуль відображення заявок співробітникам в залежності від рівня доступу.
3. Модуль API, що відповідає за зв'язок CRM з компаніями-партнерами, що надають кінцеві послуги клієнту за наданою інформацією від турагенства.
4. Модуль формування договорів та звітності для сторін оферти та бухгалтерії відповідно.

Модуль прийняття заявок складається з таких пунктів:

- введення клієнтом персональної інформації;
- перевірка внесеної інформації;
- пропонування можливих варіантів авіаквитків;
- розрахунок приблизної ціни за тур;
- виведення результатів бронювання на екран.

Модуль відображення заявок співробітникам можна розбити на наступні задачі:

- відображення заявок, що надійшли;
- корегування інформації у випадку помилкових значень;
- виведення кінцевого результату на екран.

Технологічними операціями є запити до баз даних компаній партнерів після корегування співробітником турагенства даних клієнта. Всі ці запити виконуються через API інтерфейс.

Запити до БД також відбуваються в таких задачах:

- внесення інформації про клієнта у БД турагенства;
- аналіз квартального попиту клієнтів.

Обидві ці задачі є підзадачами модулів формування звітностей та договорів.

Схематично архітектуру CRM можна зобразити наступним чином:

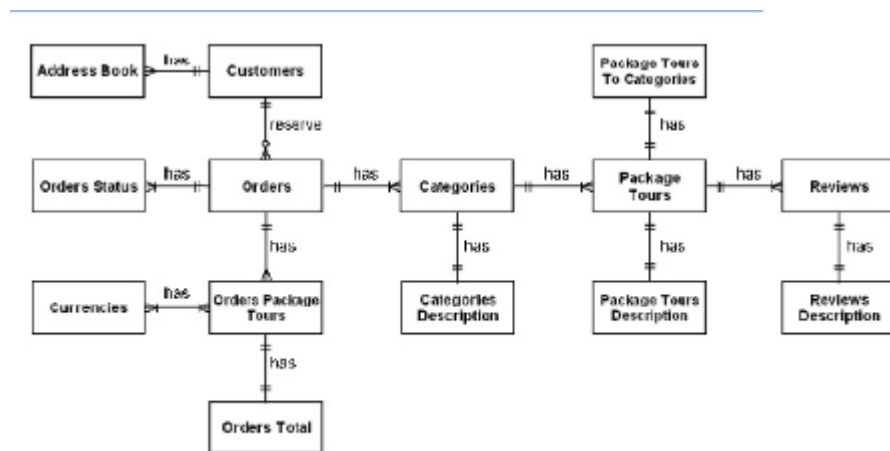


Рисунок 2.2. Архітектура CRM турагенства

3 ПРОЕКТУВАННЯ КОМПОНЕНТІВ СИСТЕМИ

3.1. Інформаційне забезпечення систем

Інформаційне забезпечення певних систем обліку має на увазі формування загального інформаційного простору, систематизацію й уніфікацію індикаторів та документів, впровадження засобів формалізованого описування відповідних даних.

Інформаційне забезпечення — це вагомий елемент автоматизованих інформаційних облікових систем, які призначено для ідентифікації інформації, що відображає стан об'єкта управління та вважається основою для ухвалення управлінських рішень.

У ході розробки інформаційного забезпечення слід зазначити:

- склад інформації, яка охоплює список інформаційних одиниць чи сукупностей, потрібних для вирішення системи завдань;
- систему інформації й її перетворення, іншими словами створення показників документів;
- показники руху інформації, іншими словами обсяг потоків, маршрути, строки;
- оцінку якості інформації;
- способи трансформації інформації.

Проведення інформаційного забезпечення проводиться поряд з певним програмним забезпеченням й інформаційною технікою, які орієнтуються на відповідного користувача. Інформаційне системне забезпечення відповідної системи можна поділити на позамашинне й внутрішньомашинне.

Загальна оцінка інформаційного забезпечення

Базою інформаційного забезпечення ІС вважається інформаційна база (ІБ), яка застосовується у функціонуванні ІС. За складом, суттю вона має відповідати потребам тих задач, планувати ті системи, що вирішуються на її базі. За сферою призначення виділяють позамашинну й внутрішньомашинну ІБ.

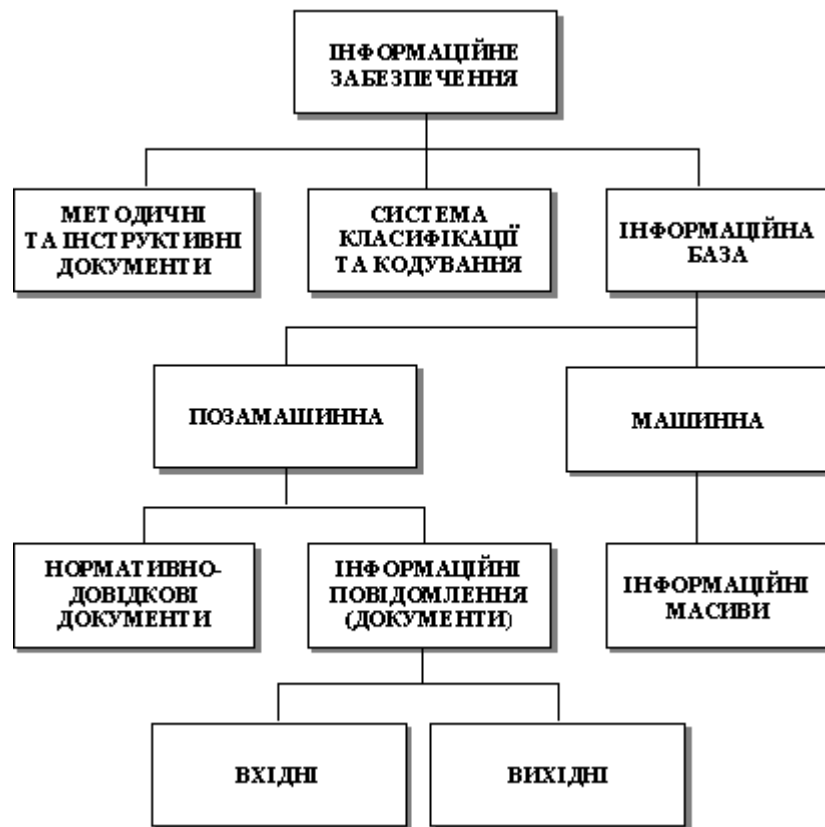


Рисунок 3.1 – Структура інформаційного забезпечення

Інформаційна база (ІБ) вважається однією з визначальних елементів інформаційного забезпечення, — тобто сукупність визначеної інформації, що застосовується в період функціонування ІС; вона та ділиться на внутрішньомашинну й зовнішньомашинну (позамашинну).

Внутрішньомашинна інформаційна база — це частина ІБ, що представляє собою систему даних на машинних носіях, застосовуваних в ІС.

Внутрішньомашинна ІБ цієї інформаційної системи створюється у формі спеціальних масивів даних, що можна умовно розділити на оперативну й нормативно-довідкову інформацію. Уся така інформація зберігається на ЖМД, проте в любий проміжок часу здатна бути заархівованою.

Зовнішньомашинна (позамашинна) інформаційна база — це частина ІБ, що представляє собою систему повідомлень, сигналів та документів, які призначені для відповідного сприйняття особою без використання засобів обчислювальної техніки. Вона вміщує вхідні, вихідні і нормативно-довідкові повідомлення.

При створенні внутрішньомашинної інформаційної бази головну увагу слід приділяти тому, щоб дані, що в ній зберігаються, можливо було широко застосовувати і щоб способи застосування даних слід було просто змінити. Для забезпечення гнучкості застосування даних, їх раціонального використання в цьому проекті створюється файлова організація внутрішньомашинної ІБ. Головною особливістю даної інформаційної бази вважається наявність нормативно-довідкових даних.

Масиви нормативно-довідкової інформації здатні застосовуватися при реалізації вводу даних, для здійснення контролю введеної інформації на наявність її у файлах НДІ.

Формування інформаційного забезпечення цієї ІС пов'язана з створенням системи класифікації й кодування, організацією збору й передачі інформації, розробки зовнішньо- та внутрішньомашинної інформаційної бази, й відповідно із створенням вихідних даних.

Формування збору та передачі первинної інформації.

Під збиранням первинної інформації розуміється фіксація інформації в місці її створення. Спосіб збирання інформації має залежність від характеристик потоків інформації, від вимог до її оперативності достовірності, від відповідних технічних засобів, які використовуються.

У ході обстеження відповідних інформаційних потоків знаходять матеріал, що характеризує управлінську та економічну роботу. В цих умовах велику увагу надають вивченню задач, які вирішуються у підрозділах об'єкта, у яких ухвалюють управлінські рішення. Сформовані інформаційні потоки повинні їх обслуговувати, тому в ході вивчення потрібно виявити цілі, критерії, завдання керування, типи цих рішень, інформаційні системи взаємодії, інформаційні джерела, її передачі, методи одержання, пошуку, зберігання.

Збір й передача інформації для обробки - це взаємопов'язані операції, оскільки вони ніколи не здійснюються окремо одна від одної.

Головними джерелами первинної інформації, що необхідно реєструвати, для формування інформаційної системи вважається фінансова звітність (Баланс й звіт про сукупний дохід).

Система керування зв'язків з клієнтами містить такі масиви даних:

Масив ORDERS – масив, у якому зберігається інформація про надходження замовлень. Спочатку додається заповнена клієнтом інформація, потім у разі потреби корегується менеджером, далі передається на наступний етап та дублюється у масив, що відповідає за зберігання історії замовлень.

У базі даних також зберігаються дані з компаній-партнерів про доступні послуги з їх сторони з дати останньої синхронізації для швидкого реагування менеджерів на пропонування існуючих варіантів і пропозицій. Дана інформація зберігається у масиві SUGGESTS та оновлюється через API інтерфейс з серверів компаній-партнерів, вона підвласна автоматичним змінам підчас запланованої синхронізації.

Джерелами для синхронізації та оновлення є:

- БД компаній-партнерів;
- клієнтські заявки;
- менеджери з замовлень;
- керівництво компанії;
- бухгалтерія.

Вимогами до організації збору інформації є:

- своєчасність та швидка заміна потрібної інформації за запитом;
- достовірність інформації;
- захищеність інформації;
- використання даних лише за призначенням;
- збереження цілісності, використовуваної інформації;
- дотримання закону про захист особистої інформації;
- дотримання положень статуту компанії про нерозголошення внутрішніх стандартів.

За своєчасне подання та зміну інформації відповідають:

- система (у випадку з заповненням клієнтом форми замовлення);
- відділ роботи з замовленнями (у разі зміни замовлення зі сторони клієнта або відмови).

Ведення кожного бізнесу своєрідно і таїть під собою труднощі, які керівникам організацій необхідно вирішувати. Готельний бізнес також ставить непрості завдання, які необхідно вирішити. Розроблена система для турагенства покликана вирішувати наступні поставлені перед нею завдання:

Автоматизація бізнес-процесів

Менеджери мають можливість вибирати конкретних виконавців і учасників поставлених завдань по протікає процесам, фіксувати час, який піде на виконання цих завдань, а керівники мають можливість аналізувати ефективність роботи співробітників, а також самі бізнес-процеси.

Управління інформацією про клієнтів

У базі даних містяться не тільки дані про клієнта, але також інформація про різних партнерів і т.д. Правильне управління цими даними дозволяє використовувати їх з максимальною ефективністю. Клієнтська база консолідована, компанія має повну картину того, що вважають за краще клієнти, і, ґрунтуючись на цьому, будує подальший сценарій взаємодії.

Керування продажами

Автоматизація виконання замовлень. Цей елемент інформаційної системи відповідає за отримання обслуговування клієнтом необхідної якості у встановлений час. Крім того, є можливість відслідковувати всі заявки клієнта, формувати документи про всі дії, які супроводжують продаж, і продумувати подальші дії по стосункам з клієнтом.

Модулі, сервіси, контролери. Їх взаємодія і зв'язку

Вся структура розроблюваної інформаційної системи може бути представлений набором модулів. Є найголовніший модуль, до якого йдуть всі зв'язки. Він відповідає за обмін даними між іншими модулями системи. Кожен модуль, який підключається до основного, реалізує свій бізнес-процес або функціонал системи. Виходить, що якщо виникне питання про те, щоб

оптимізувати систему, то відповідь проста - достатньо лише розробити модуль для виконання поставленого завдання і підключити його до основного модуля CRM.

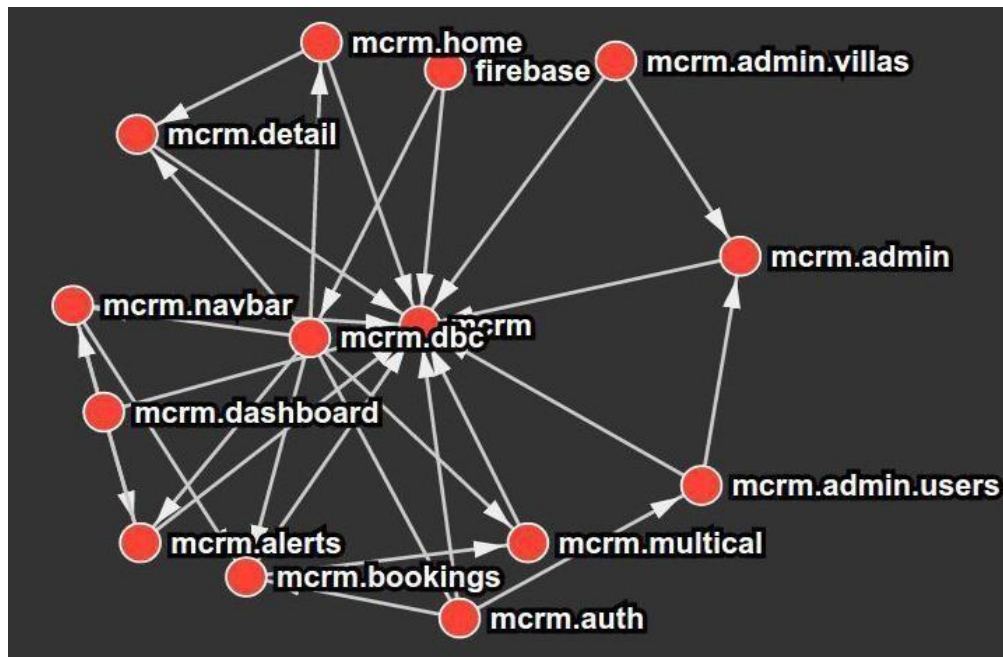


Рисунок 3.2 – Залежності модулів CRM

В CRM входять:

- Основний модуль - mcrm. Він запускає роботу системи і до нього підключаються всі інші модулі.
- Модуль роботи з базою даних - mcrm.dbc
- Модуль навігації по системі - mcrm.navbar
- Модуль домашньої сторінки - mcrm.home
- Модуль з інструментами для роботи з базою даних - firebase
- Модуль відображає детальну інформацію про приміщення
 - - mcrm.detail
- Модуль налаштування системи, що містить системну інформацію - mcrm.dashboard
 - Модуль бронювання - mcrm.bookings
 - Модуль реєстрації / авторизації - mcrm.auth

- Модуль логування - mcrm.alerts
- Модуль мультікалендаря - mcrm.multical
- Модуль адміністратора - mcrm.admin, який включає в себе підмодулі управління приміщеннями - mcrm.admin.villas і управління користувачами - mcrm.admin.users

Всі модулі містять у собі логіку виконуваної ними процесу, а для виконання цих процесів використовуються сервіси. Їх зв'язок відображений на рисунку 3.3.

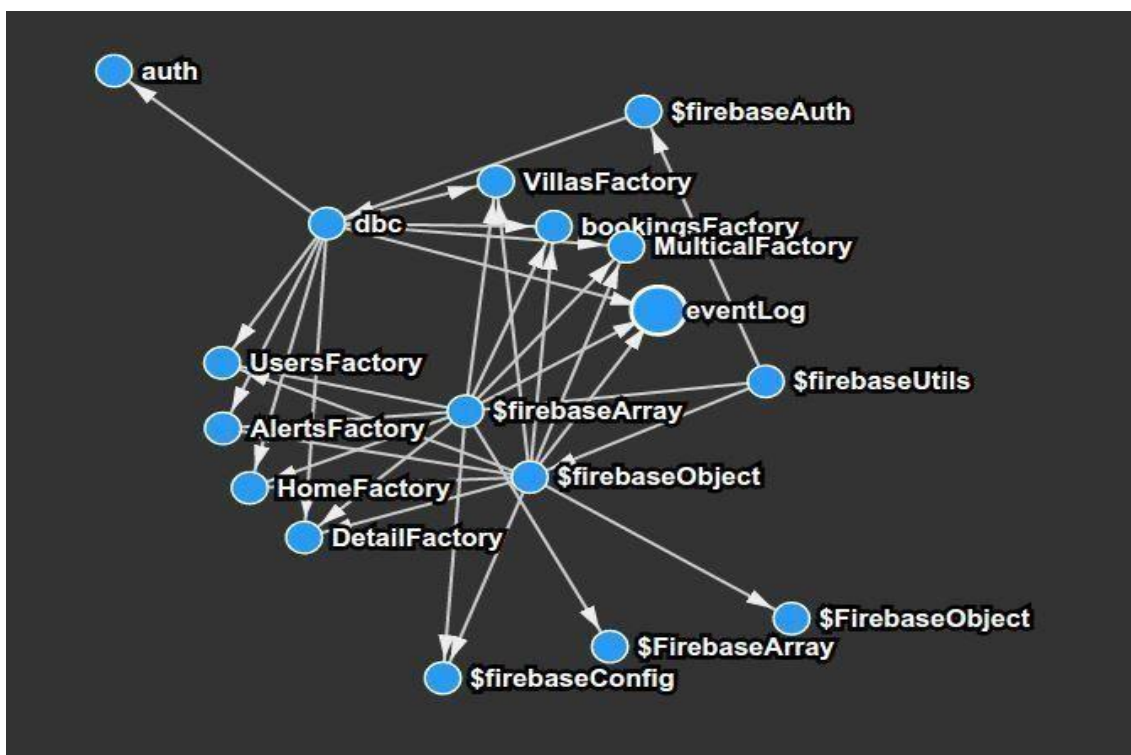


Рисунок 3.3 – Залежність сервісів CRM

Вся логіка модулів описана в контролерах. Сервіси є сінгтонами, які є шаблонами проектування, що обмежують клас таким чином, щоб у нього був тільки один екземпляр.

Це означає, що до сервісів можна звернутися з будь-якого модуля додатку. При звертанні до сервісів можна отримати різні дані з інших модулів або виконати необхідні дії для здійснення процесу, скориставшись тим чи іншим сервісом. Залежність контролерів і сервісів можна побачити на рис. 3.4.

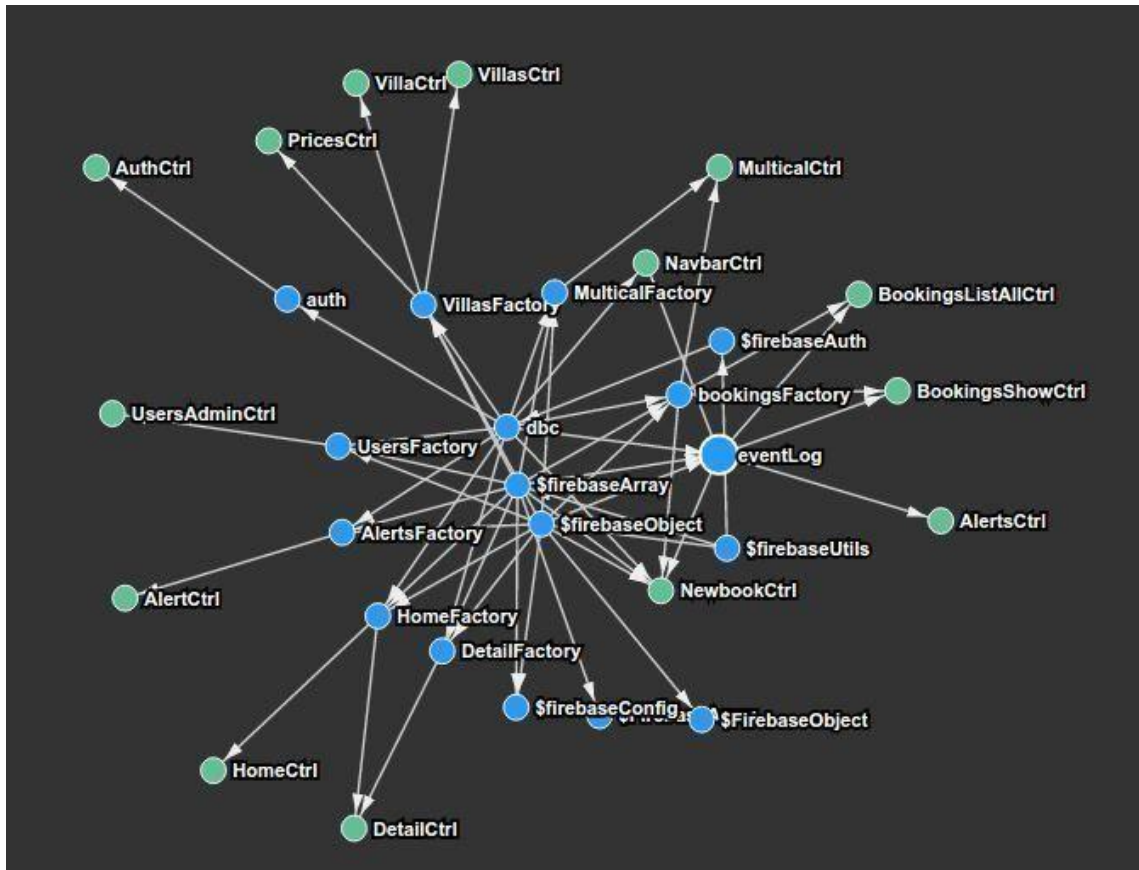


Рисунок 3.4 – Залежність контролерів і сервісів CRM

Ось приклад роботи одного з модулів - `crm.home`. Логіку модуля описує контролер `HomeCtrl`. Для виведення даних домашньої сторінки їх потрібно спершу взяти з бази даних. За це взаємодія відповідає сервіс - `dbc`. Контролер використовує сервіс - `HomeFactory`, який, використовуючи `dbc`, отримує дані з бази даних. В кінцевому підсумку `HomeCtrl` отримує дані з сервісу `HomeFactory`, які вже потім слідують на домашню сторінку програми.

Сутності

Всебічне вивчення об'єктів автоматизації, її властивостей, взаємовідносини носіння між цими об'єктами і представлення отриманої інформації у вигляді моделі даних є одним з ключових моментів створення інформаційної системи. У цьому випадку проводиться концептуальне проектування.

Концептуальна модель призначена для подання семантики предметної області в термінах суб'єктивних засобів опису (сутностей, атрибутів, ідентифікаторів сутностей) і містить наступні основні конструкції:

- Моделі «сутність-зв'язок» (ER).
- Визначення сутностей.
- Визначення атрибутів сутностей.
- Унікальні ключі-ідентифікатори.
- Відносини між сутностями.

Модель «сутність-зв'язок» представляє собою графічну нотацію, яка ґрунтується на блоках і сполучних лініях. За їх допомогою описуються об'єкти моделі і відносини між ними. ER має кілька основних понять:

- Сутність - це реальний чи уявлений об'єкт, інформація щодо якого має зберігатися та бути доступною. У графіках ER- моделі зміст представляється у формі прямокутника, вміщуючого ім'я сутності. Водночас, ім'я сутності - це ім'я типу, а не певного конкретного екземпляра даного типу.

- Атрибутом сутності вважається будь-яка деталь, що слугує для уточнення, ідентифікації, класифікації, цифрової характеристики чи відображення стану сутності. Імена атрибутів вносяться в прямокутник, який відображує сутність, під ім'ям сутності та відображуються малими літерами, можливо, з прикладами.

- Зв'язок - бінарна асоціація двох і більше сутностей, тобто вона показувала, як пов'язані екземпляри сутностей між собою. Зв'язки діляться на три типи: «один-до-одного» (1: 1), «один-до-багатьох» (1: M), суті з одним екземпляром іншої сутності. 1: M означає зв'язок однієї сутності з кількома екземплярами іншої. M: M означає зв'язок багатьох сутностей до багатьох.

У розробляється інформаційній системі виділені сутності:

- Бронювання «booking». Містить в собі всю інформацію про поступили заявки на бронювання.

- Зайнятості готелю «busy». Містить в собі дати бронювання по- місцень готелю.

- Інформації про приміщення «villas». Містить опис приміщення.
- Користувачі «users». Містить інформацію про зареєстрованих користувачів.

- Повідомлення «event Log». Містить інформацію про будь-які зміни.
- Вид події «action». Дії про які приходять повідомлення.
- Займані приміщення «villa».

Для сутності «bookings» виділено такі атрибути:

- Ідентифікаційний номер (ключ)
- Поштова адреса користувача
- Ім'я
- Прізвище
- кількість персон
- Телефон
- Дата початку бронювання
- Дата кінця бронювання
- Час початку
- Ідентифікаційний номер (сніданки)
- Ідентифікаційний номер (вілли)
- Список займаних приміщень (вілла)

Для сутності «busy» виділено такі атрибути:

- Ідентифікаційний ключ (дата)
- Ідентифікаційний ключ (villas)
- Ідентифікаційний ключ (bookings)

Для сутності «villas» виділено такі атрибути:

- Ідентифікаційний номер (ключ)
- Адреса
- кількість спалень
- ім'я
- Ціна за ніч

- опис

Для сутності «users» виділено такі атрибути:

- Ідентифікаційний номер (ключ)
- Поштова адреса
- ім'я
- Роль користувача в системі
- реєстраційний ключ

Для сутності «event Log» виділено такі атрибути:

- Ідентифікаційний номер (ключ)
- Дата події
- час події
- Ключ об'єкта події
- Дія
- Ідентифікаційний ключ (users)

Для сутності «action» виділено такі атрибути:

- Ідентифікаційний номер (ключ)
- надходження заявки
- зміна заявки
- підтвердження заявки
- відхилення заявки
- Додавання приміщення

Для сутності «villa» виділено такі атрибути:

- ім'я приміщення
- Ідентифікаційний ключ (villas)

У концептуальній моделі мають місце такі зв'язки:

- між сутностями «bookings» і «busy» встановлюються відношення М: М, так як багато заявок на бронювання і кожною заявкою відповідає своя запис по сутності «busy».

- між сутностями «bookings» і «villa» встановлюються відношення 1: M, так як в одній заявці може бути заброньовано одне або кілька приміщень.
- між сутностями «busy» та «villas» встановлюються відношення M: M, так як може бути заброньовано кілька приміщень і кожному з приміщень відповідає своя запис по сутності «Villas».
- між сутностями «villa» і «villas» встановлюються відношення M: M, так як може бути заброньовано кілька приміщень і кожному з приміщень відповідає своя запис по сутності «Villas».
- між сутностями «event Log» і «bookings» встановлюються відношення M: M, так як додавання, зміна сутності «Bookings» відповідає екземпляр сутності «event Log».
- між сутностями «event Log» і «users» встановлюються відношення 1: 1, так як повідомлення з'являється у відповідь на дії ка- кого-небудь користувача і цей користувач один

У результаті була отримана структура бази даних, представлена на рисунку 3.5.

Опис функціоналу:

CRM турагенства вдає із себе програмний комплекс для автоматизації роботи туристичного агентства.

Права і можливості груп користувачів:

Власник може:

- CRUD даних про доступні для брон.
- CRUD користувачів CRM.
- Переглядати статистику і звіти CRM.

Менеджер може:

- CRUD броні.
- Підтверджувати / Відхилити / Пропонувати варіанти на запити на бронь.
- Підтверджувати запити на бронь від гостей.

Клієнт може:

- Переглядати доступність вілли для конкретної дати, діапазону дат.

- Залишити заявку на бронювання.
- Переглядати дані про віллу.

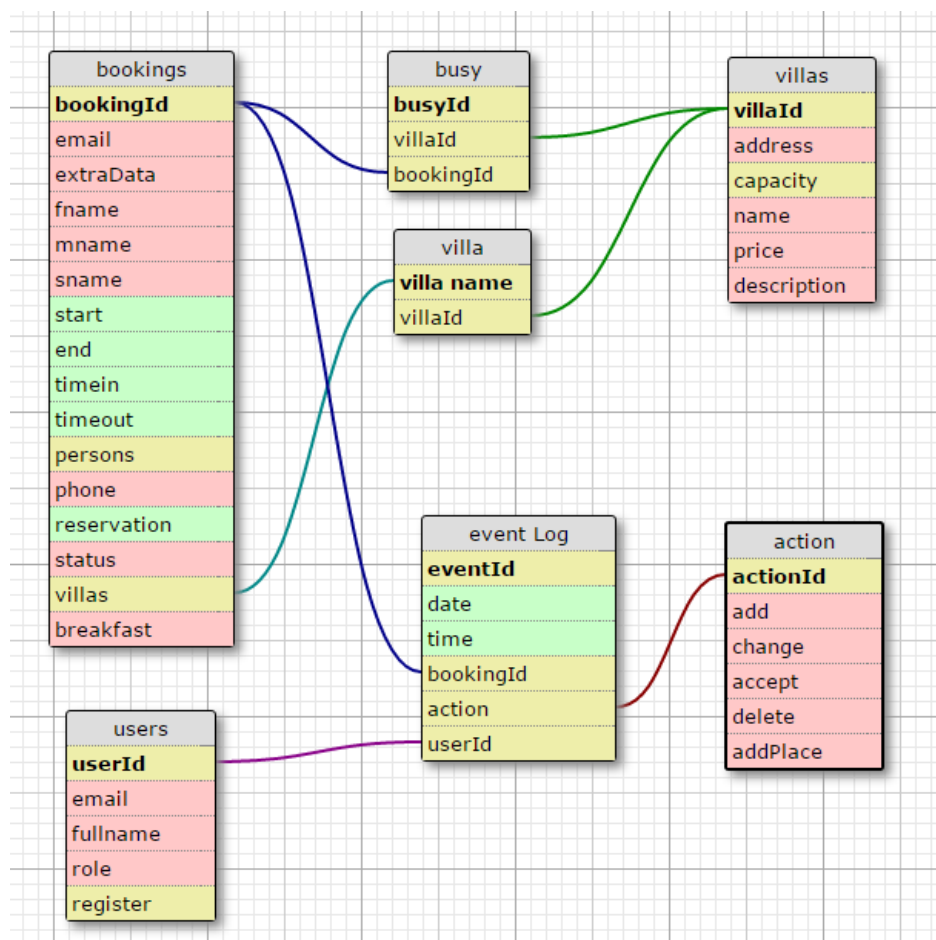


Рисунок 3.5 – Модель БД

Головна сторінка:

На головній сторінці зображується шахматка зайнятих номерів в готелі, яка представляє собою сітку з датами і номерами в готелі, на якій наступними кольорами виділені: блакитний - номер вже зайнятий проживають, сірий - номер, з якого виїхали, і зелений - майбутні заїзди в номери готелю . Шахматка на головній сторінці розташовується в першу чергу для персоналу, щоб працівники бачили завантаженість готелю. Які номери зайняті, а які ні.

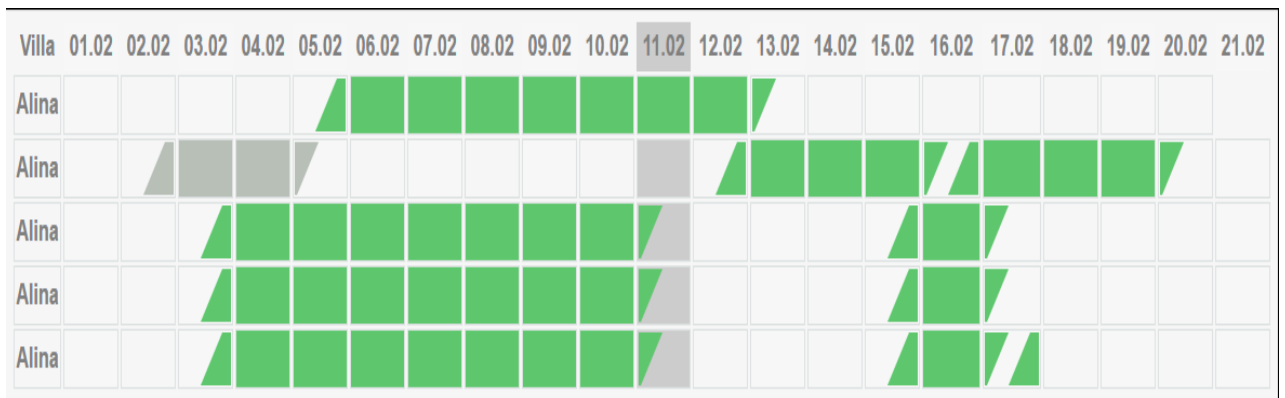


Рисунок 3.6. – Шахматка зайнятих номерів в готелі

Деталі з бронювання

Елементи управління:

Delete in cal - видалити з Google-календаря і перенести подія в віддалені. З віддалених подія може бути відновлена шляхом редагування і синхронізації, або видалено остаточно.

Sync - збереження і синхронізація події з Google-календарем.

Print / Print contract - друк і друк договору про прийнятих умовах.

Denial - скасування внесених змін.

У даних про броні обов'язково фіксується дата і час заносу броні для того, щоб можна було спостерігати статистику і мати відомості в яку дату і в який час клієнт бронює номер.

Управління маркетингом

Цей елемент головним чином спрямований на те, щоб автоматизувати роботу з клієнтом, здійснювати підтримку цієї роботи, а також сприяє розвитку взаємин про допомоги проведення різних досліджень і подальшого аналізу отриманих даних. Здійснюється реєстрація потенційних клієнтів, виявлення їх потреб, а також формування пропозицій.

Управління взаємодіями

Збір і управління загальною інформацією про клієнтів, як уже існуючих, так і потенційних. До такої інформації відносяться контакти, історія замовлень, історія взаємодій і зворотний зв'язок. Можливий процес переселення з одного готелю в інший.

Веб-аналітика

Збір, аналіз та інтерпретація інформації про відвідувачів веб-сайту. Підготовка звітів по зайнятості готелів. Можливість підвищити рентабельність готелів, провести аналіз системи продажів, що необхідно для прийняття подальших рішень.

Управління персоналом

Розробка способів максимальної ефективності процесів використання, розвитку і мотивації своїх співробітників з метою отримання максимальної задоволеності клієнта, а також для досягнення поставлених цілей комплексом вілл. Постановка завдань для персоналу. Можливість співробітникам відзначати отримані результати при виконанні запланованих справ, переглядати зміни, які вносив кожен працівник.

Інтеграція

Обмін інформацією між своїми модулями і зовнішніми системами, наприклад, з такими, як календар Google і booking.com.

Представлення системи

Підчас входу до системи керування з акаунту адміністратора ми бачимо привітливий надпис:

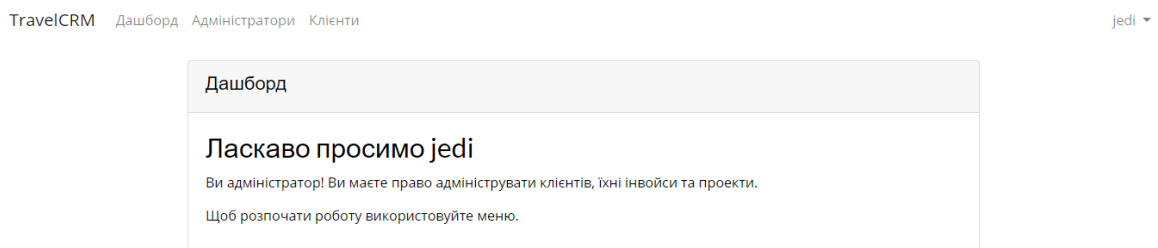


Рисунок 3.7 – Головна сторінка CRM

Тут представлено вікно керування адміністрацією та персоналом:

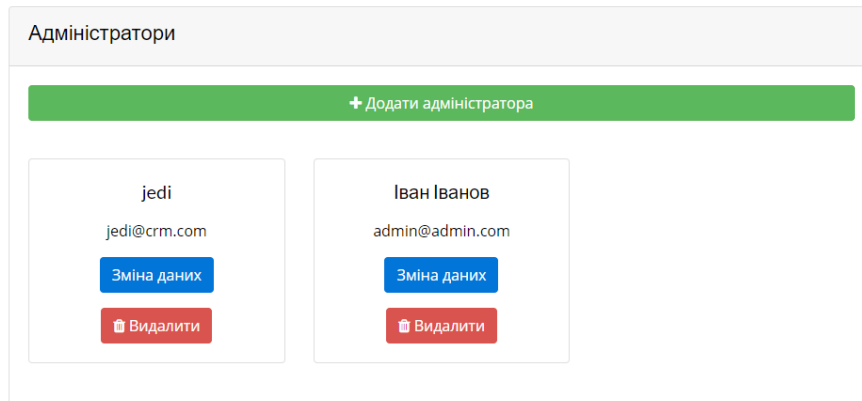


Рисунок 3.8 – Вікно керування доступом до системи

Модальне вікно редагування користувача системи:

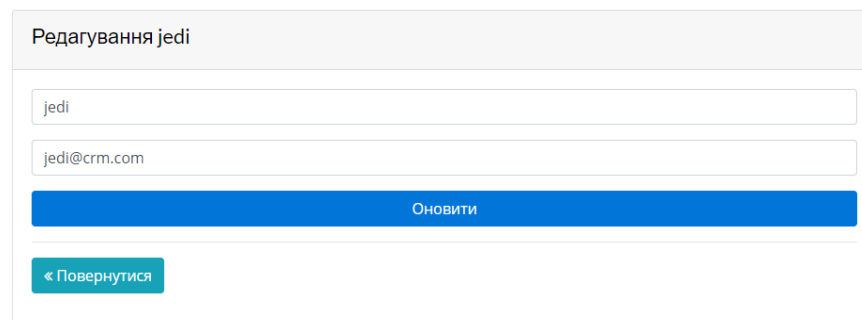


Рисунок 3.9 – Редагування головного адміністратора

Панель перегляду нових заявок від клієнтів:

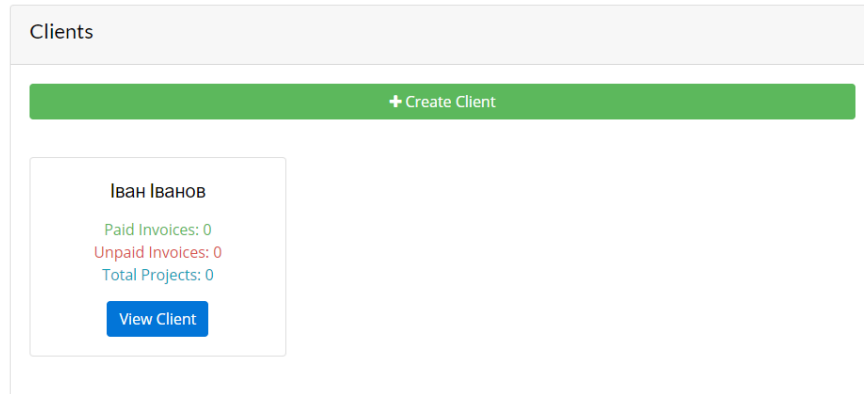


Рисунок 3.10 – Вікно перегляду клієнтів системи

Панель перегляду стану заявок від клієнта та його контактні дані:

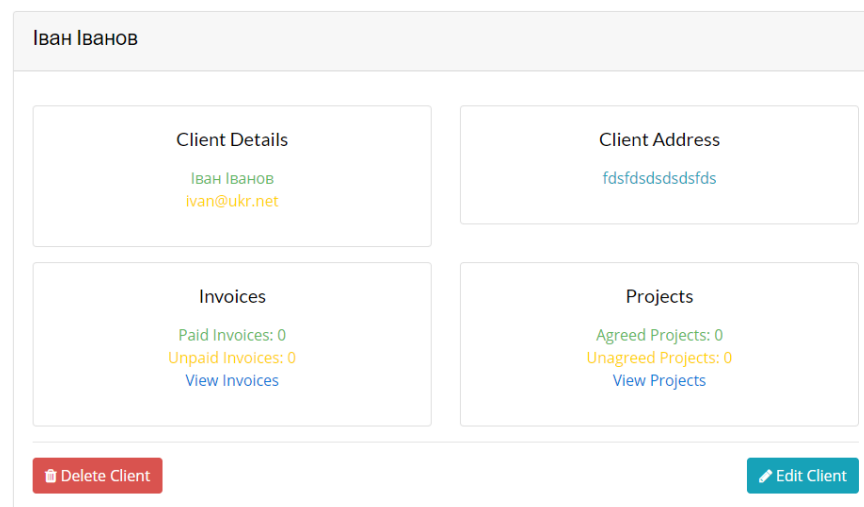
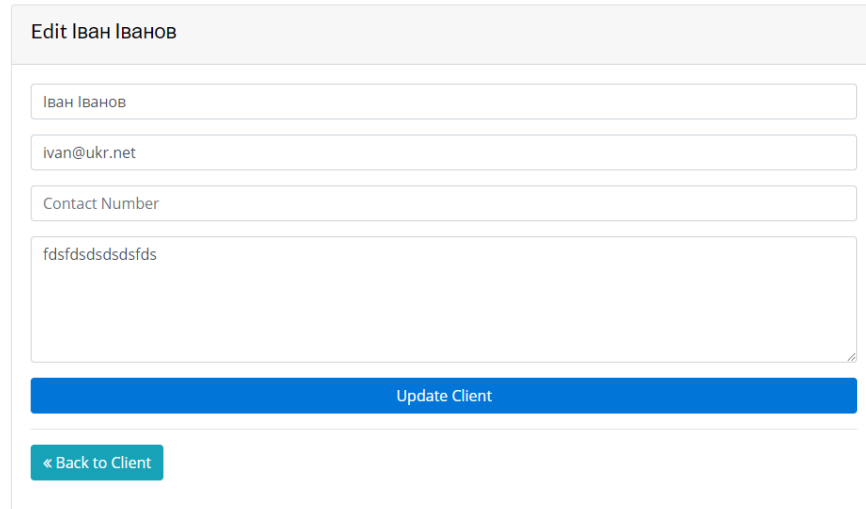


Рисунок 3.11 – Вікно перегляду стану заявок від клієнта

Модальне вікно керування обліковими даними клієнта:

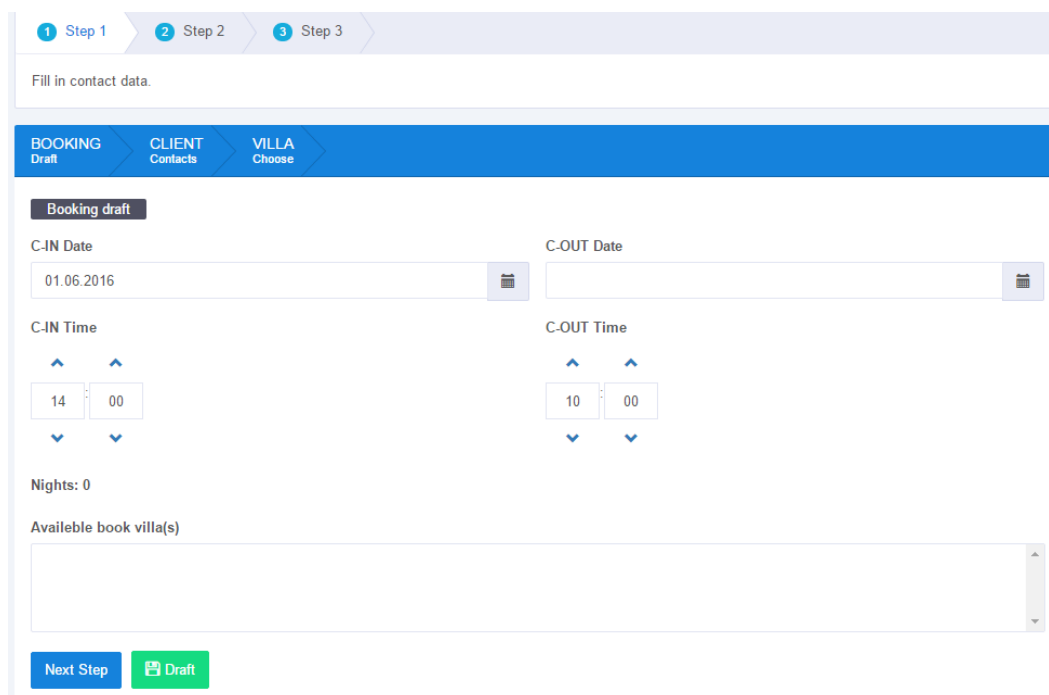


The screenshot shows a web form titled "Edit Іван Іванов". It contains several input fields: a text field with "Іван Іванов", an email field with "ivan@ukr.net", a "Contact Number" field, and a larger text area containing "fdsfsdsdsdsfds". At the bottom, there is a blue "Update Client" button and a teal "← Back to Client" button.

Рисунок 3.12 – Вікно керування клієнтом

Дані про броні

Заповнюються поля FirstName (ім'я), LastName (прізвище), Villa (вибір тура), Telephone (телефон), Email (електронна пошта), Data check in (дата в'їзду), Data check out (дата виїзду), показ кількості ночей поруч, скільки людина в'їжджає, сніданок так / ні, паспортні дані, оплата і іншу маячню. Вся послідовність дій при заповненні даних про бронювання по кроках відображена на наступних рисунках:



The screenshot shows a multi-step booking process. At the top, there are three steps: "Step 1" (active), "Step 2", and "Step 3". Below the steps, it says "Fill in contact data." There is a navigation bar with three tabs: "BOOKING Draft", "CLIENT Contacts", and "VILLA Choose". Underneath, there is a "Booking draft" section. It contains fields for "C-IN Date" (01.06.2016) and "C-OUT Date", "C-IN Time" (14:00) and "C-OUT Time" (10:00). Below these is "Nights: 0" and a scrollable area for "Available book villa(s)". At the bottom, there are "Next Step" and "Draft" buttons.

Рисунок 3.13 – Заповнення даних броні (крок 1)

1 Step 1 2 Step 2 3 Step 3

Choose villa.

BOOKING Draft CLIENT Contacts VILLA Choose

Booking Data

First Name

Last Name

Middle Name

Phone Name

Email Name

Previous Step Next Step

Рисунок 3.14 – Заповнення даних броні (крок 2)

BOOKING Draft CLIENT Contacts VILLA Choose

Villa Data

C-IN Date: 01.06.2016

C-OUT Date:

C-IN Time: 14:00

C-OUT Time: 10:00

Nights: 0

Available book villa(s)

Number of persons: 1

Breakfast:
 Yes No

Number of breakfast: 0

Previous Step Send

Рисунок 3.15 – Заповнення даних броні (крок 3)

Майстер з бронювання.

У майстра з бронювання заповнюються дані по броні в покроковому режимі. Недоступно подвійне бронювання або занесення броні на тур, якщо вже є пересічна бронь.

Формування check in / check out

На цій сторінці можна вносити дані про водія, який зустрічає гостя, або іншої відповідальної особи, час чек-ін та чек-ауту, а також дані про приліт і конкретної броні.

Постояльці готелів

Відображення готелю, початок і закінчення броні, дані про мешканців готелю, і, якщо такий є, залишок боргу.

Найближчі броні

Висновок даних про найближчу бронь в зручному вигляді.

Повідомлення

У висувній панелі прийняті наступні позначення для скорочень:

B (booking) - була зроблена бронь.

G (google) - була змінена броні в Google.

Z (клієнт) - надійшов запит на бронь від клієнта або менеджера.

3.2. Технічне забезпечення

Технічне забезпечення відповідних інформаційних систем – це система певних технічних засобів, до якої належать засоби обчислювальної техніки, устаткування для формування локальних мереж та підключення до глобальних мереж, засоби реєстрації, відображення інформації та її накопичення, створені для автоматизованого формування, накопичення, переробки, передачі, обміну й відображення інформації.

До головних компонентів системи технічного забезпечення відповідних інформаційних систем відносять технічні засоби обчислення; периферійне устаткування (обладнання для накопичення та зберігання даних: засоби транслітерації інформаційних джерел); пристрої для автоматичного зчитування даних; торговельне приладдя (сканери, комп'ютерні касові апарати); засоби керування технологічними й виробничими процесами; пристрої для комунікацій; обладнання для передавання й обміну даними; мережеве устаткування.

Формування системи обробки даних й інформаційних технологій ґрунтується на загальних засадах виявлення й вибору конфігурації системи технічних засобів, які створюють раціональне інформаційне обслуговування відповідних рівнів та об'єктів керування, а відповідно враховують фактори виконання функцій та процедур керування в інформаційній системі. Кількість та склад застосовуваних технічних засобів характеризуються обсягами й інтенсивністю інформаційних потоків, технологією виконання функцій керування та їх терміновими особливостями. Підвищення вимог до економічної інформації й покращення управління виявляють потребу використання продуктивніших та розвиненіших засобів і технологій.

Загальні положення.

Початковими технічними засобами для створення даної інформаційної системи були:

- персональний комп'ютер;
- веб сервер для тестування та налагодження серверної частини;

Для коректної роботи аплікації потрібно:

- Персональний комп'ютер чи смартфон з доступом в інтернет;
- прийнятна швидкість інтернету не менше 2мбіт/сек;
- Налаштований веб сервер на основі ОС Linux.

Структура комплексу технічних засобів.

При виборі технічних засобів потрібно керуватися такими критеріями:

- забезпечення достатньої обчислювальною потужністю;
- Веб сервер Apache чи nginx, бажано у поєднанні;
- остання версія веб браузеру;

Мережна апаратура передачі даних

Для функціонування системи потрібно мережеве обладнання як зі сторони сервера так і зі сторони клієнта. Веб сервер та клієнт повинні бути забезпечені підключенням до мережі інтернет. У нашому модельному випадку сервер підключений напряду через порт Ethernet, менеджер використовує офісний ПК з підключенням через коаксіальний кабель (DOCSIS 3.0) протокол.

Підключення іншим чином для адміністраторів неможливе так як закладена апаратна прив'язка заради безпеки системи.

Клієнт-замовник може використовувати будь-яке підключення.

3.3. Програмне забезпечення

Під програмним забезпеченням (ПЗ) інформаційних систем слід розуміти сукупність відповідних за функціями, взаємозалежних програмних та документальних засобів, які створюють та експлуатують системи обробки даних.

Структура програмного забезпечення.

Структура ПЗ представлена на рис. 3.16.

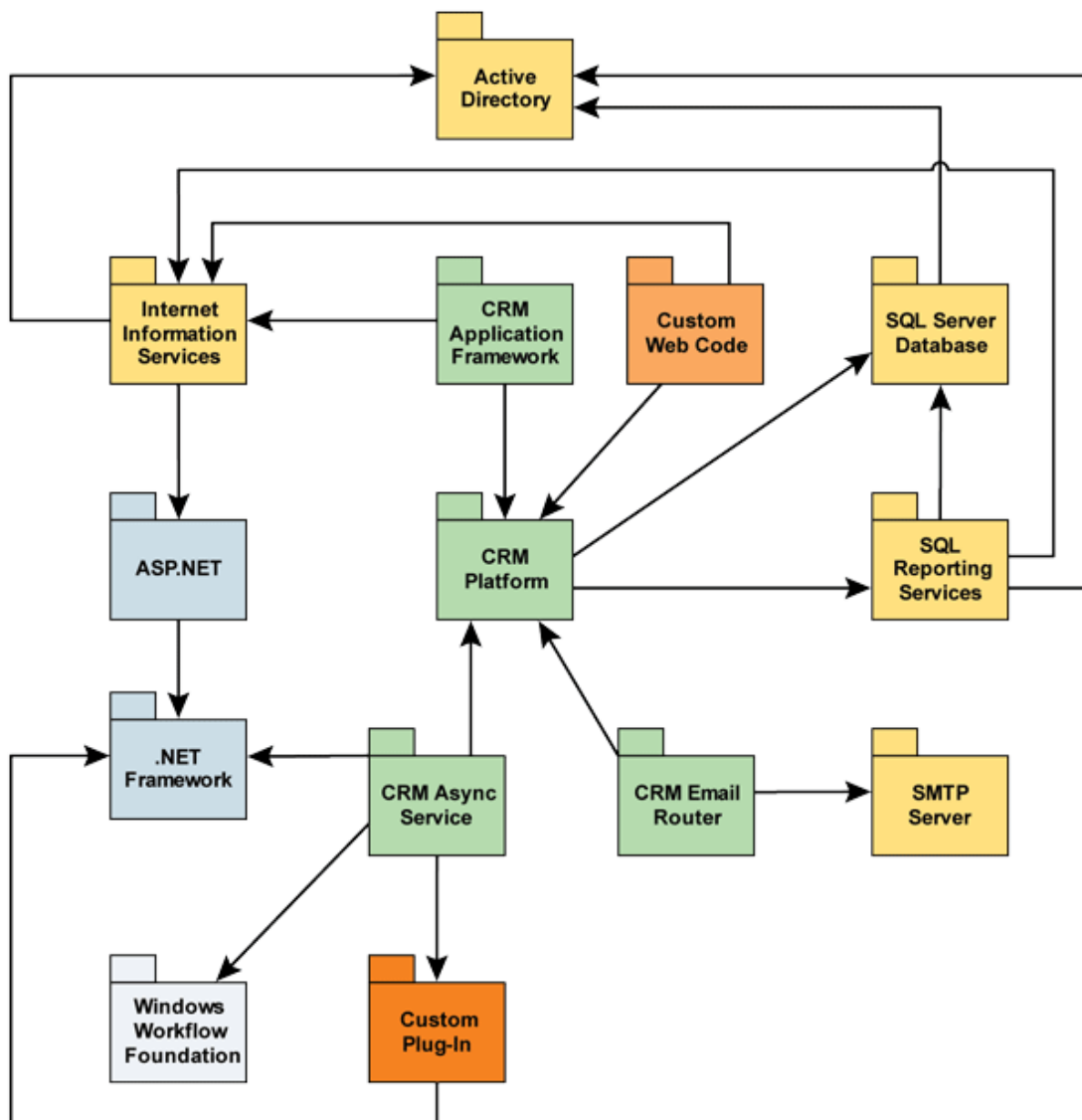


Рисунок 3.16 – Структура програмного забезпечення

Системне програмне забезпечення

Опишемо системне програмне забезпечення по пунктам [8-10].

Apache можна представити як HTTP веб-сервер. Веб сервер Apache – це кросплатформенне програмне забезпечення. На базі вагової документації та відповідної інтеграції зі стороннім ПЗ, Apache отримав значне розповсюдження. Підтримує такі ОС - BSD, Linux, Microsoft Windows, Mac OS, NetWare, BeOS, Novell.

Перевагами веб-сервера Apache є наступні: підтримка таких мов програмування, як Ruby, PHP, Python, ASP, Perl, Tcl; простота у підключенні зовнішніх систем; підтримка систем PHP, CGI і FastCGI; існування механізмів, що забезпечують чіткість та розмежування доступу інформації; здатність застосовувати СУБД з метою аутентифікації користувачів; гнучка та потужна конфігурація системи; сумісна з додатками, що потребують потужного криптографічного захисту інформації; здатність формування користувацьких директорій для веб-сайту; здатність створення віртуальних хостів, на основі яких на одному фізичному сервері можливо відтворити декілька віртуальних; відтворює протоколи тих подій, які відбуваються на зовнішньому сервері; активний зворотний зв'язок з розробниками та вчасне усунення виниклих помилок в ПЗ.

Обмін інформацією в мережі Інтернет відбувається з використанням протоколів прикладного рівня, які створюють той чи інший практичний сервіс (пересилання файлів, гіпертекстових даних, пошта тощо).

Одним з наймолодших й дуже популярних сервісів системи Інтернет, розвиток якого спричинив до зростання популярності власне Інтернет, слід вважати World Wide Web (WWW), який ґрунтується на протоколі HTTP (Hypertext Transfer Protocol – це протокол передачі гіпертекстової інформації). Гіпертекстові документи, що є представленими в WWW, характеризуються однією принциповою особливістю поряд з традиційними гіпертекстовими документами - зв'язки, в них які використовуються, не обмежені відповідним документом, та також, не обмежені певним комп'ютером. З ціллю формування

гіпертекстових документів застосовується мова HTML (Hyper Text Markup Language – мова для розмітки гіпертекстових даних), яка використовує вагомі здатності щодо редагування й структурної побудови документів, організації зв'язків між відповідними документами, засоби включення графічної та мультимедійної інформації. HTML-документи вважаються видимими на основі спеціальної програми, а саме браузера. Найбільш поширеними нині є браузери Mozilla Firefox та Internet Explorer фірми Microsoft (MSIE). Програми Mozilla Firefox доступні майже для всіх провідних апаратних та програмних платформ, діяльності MSIE доступні для провідних Windows платформ, Macintosh та окремих комерційних Unix-систем.

HTML-документ містить текст, який є змістом документа, і тег і в , що формують його структуру та зовнішній інтерфейс при відображенні браузером.

Простий html-документ виглядає таким чином:

```
<html >
<head >
<title>Назва</title>
</head>
<body >
<p>Основа документу
</body>
</html>
```

Як зрозуміло з прикладу, тег вважається ключовим символом, поміщеним у кутові дужки.

Відокремлюють одинарні теги, такі як, зокрема, <p > , та парні, такі як <body></body>, в цьому випадку дія тега розповсюджується лише на текст поміж його відкриваючою та закриваючою дужкою. Також, теги здатні мати параметри - наприклад, при описі сторінки можна задати колір фону, колір шрифту і т.д.: <Body bgcolor="white" text="black">.

Текст відповідного документа заключається в теги <html > , власне документ розбивається на певні частини - заголовок та тіло. Заголовок має опис

за тегами <head>, де можуть включатись назва документа (на основі тегов <title>) та інші параметри, які застосовуються браузером при створенні документа. Тіло документа вміщується в теги <body> та вміщує власне дані, які спостерігає користувач. При відсутності тегів форматування всього тексту трансформується у вікно браузера суцільним потоком, переклади рядків, пропуски та табуляції ідентифікуються в якості пробільних символів, декілька пробільних символів, які розташовуються підряд, замінюються на один. З метою форматування застосовуються наступні головні теги:

<p> - початок нового абзацу, матиме параметр, який характеризує вирівнювання:

<p align=right>;

 - переклад рядка в межах поточного абзацу;

<u></u> - виділення тексту підкресленням

Посилання на інший документ формується на основі тега ..., де URL – це повна або відносна адреса документа. За таких умов текст, ув'язнений в тег <a>, звичайно виокремлюється підкресленням та кольором, й після натискання мишею за цим посиланням браузер відкриває документ, адреса якого відображена за параметром href. Графічні зображення вставляються в документ на основі тега .

Нині вже всім спеціалістам в сфері веб-технологій стає очевидніше, що діючих стандартів передавання даних по Internet стає недостатньо. Формат HTML, ставши колись революцією в галузі відображення вмісту вузлів Internet, нині не задовольняє всім потрібним на цей момент вимогам. Він дає можливість описати те, як саме мають бути відображені дані на екрані кінцевого користувача, проте не створює ніяких засобів для раціонального опису даних, що передаються, та управління ними.

Також, проблемою для численних компаній, що здійснюють розробку програмного забезпечення, вважається потреба у сумісному застосуванні різних компонентів, забезпечення їх взаємодії, можливості обміну даними між ними.

Ще не так давно не було стандарту, який надає засоби для інтелектуального пошуку інформації, обміну даними, адаптивної обробки держання даних.

Вирішенням цих проблем став затверджений в 1998 році міжнародною організацією W3C мова XML. XML (eXtensible Markup Language) - це розширювана мова розмітки, призначена для опису в текстовій формі структурованих даних. Цей текстовий (text-based) формат, багато в чому схожий з HTML, розроблений спеціально для зберігання і передачі даних.

XML дозволяє описувати і передавати такі структуровані дані, як:

1. Окремі документи.
2. Метадані, що описують вміст якого-небудь вузла Internet.
3. Об'єкти, що містять дані і методи роботи з ними (наприклад, елементи управління ACTIVEX або об'єкти Java).
4. Окремі записи (наприклад, результати виконання запитів до баз даних)
5. Всілякі веб-посилання на інформаційні і людські ресурси Internet (адреси електронної пошти, гіпертекстові посилання і ін.)

Дані, описані на мові XML, називаються XML-документами. Мова XML легко читана і достатньо проста для розуміння. Якщо Ви були знайомі з HTML, то навчитися складати XML-документи не складе для Вас ніяких труднощів.

Початковий текст XML-документа складається з набору XML-елементів, кожен з яких містить початковий і кінцевий теги. Кожна пара тегів представляє частину даних. Тобто, як і HTML, мова XML для опису даних використовує теги. Але, на відміну від HTML, XML дозволяє використовувати необмежений набір пар тегів, кожна з яких представляє не то, як ув'язнені в неї дані повинні виглядати, а то, що вони означають.

Любий елемент XML-документа здатен мати атрибути, характерні риси, які уточнюють його зміст. Атрибут - це пара ім'я = "значення", що закладається при створенні елементу в першому тегу.

Принцип розширюваності мови XML полягає в можливості використання необмеженої кількості пар тегів, визначуваних творцем XML-документа.

Принцип незалежності визначення внутрішньої структури документа від способів представлення цієї інформації визначається виділенням даних від процесу їх обробки та відображення. Отже, одержані дані слід застосовувати відповідно до вимог клієнта, іншими словами вибирати необхідне оформлення, використовувати потрібні методи обробки.

Керувати відображенням елементів у вікні програми-клієнта (наприклад, у вікні браузера) можливо на основі спеціальних інструкцій - стилєвих таблиць XSL (eXtensible Stylesheet Language). Такі таблиці XSL дають можливість сформуванню оформлення елемента залежно від його.

```
protected $fillable = [
    'name', 'email', 'password', 'role', 'is_admin', 'id',
];

/**
 * The attributes that should be hidden for arrays.
 *
 * @var array
 */
protected $hidden = [
    'password', 'remember_token',
];

/**
 * Returns true if the user is an admin and
 * false if not.
 *
 * @return bool
 */
public function isAdmin()
{
    return $this->is_admin;
}

public function isSuperAdmin()
{
    if ($this->id === 1) {
        return true;
    } else {
        return false;
    }
}

public function client()
{
    return $this->hasOne(Client::class);
}
```

місцерозташування усередині документа, тобто до двох елементів з однаковою назвою можуть застосовуватися різні правила форматування. Крім того, мовою, лежачою в основі XSL, є XML, а це означає, що таблиці XSL більш універсальні, а для контролю коректності складання таких стильових таблиць можна використовувати DTD-опису або схеми даних, розглянуті нижче.

Формат XML, в порівнянні з HTML, має невеликий набір простих правил розбору, який дозволяє розбирати XML-документи, не вдаючись до яких-небудь зовнішніх описів використовуваних XML-елементів. У загальному випадку XML-документи повинні задовольняти наступним вимогам:

1. Кожен відкриваючий тег, що визначає деяку частину даних в документі, обов'язково повинен супроводжуватися таким, що закриває, тобто, на відміну від HTML, не можна опускати закриваючі теги.

2. Вкладеність тегів в XML строго контролюється, тому необхідно стежити за порядком проходження відкриваючих і закриваючих тегів.

3. У XML враховується регістр символів.

4. Вся інформація, розташована між початковим і кінцевим тегами, розглядається в XML як дані, і тому враховуються всі символи форматування (тобто пропуски, переклади рядків, табуляції не ігноруються, як в HTML).

У XML існує набір зарезервованих символів, які повинні бути задані в XML-документи тільки спеціальним чином. Багато фахівців розглядають XML як нову технологію інтеграції програмних компонентів. Основними перевагами використання XML є:

1. Інтеграція даних з різних джерел. XML можна використовувати для об'єднання різнорідних структурованих даних на середньому рівні трирівневих веб-систем, баз даних.

2. Локальна обробка даних. Отримані дані у форматі XML можна розбирати, обробляти і відображати безпосередньо на клієнті.

3. Перегляд і маніпулювання даними в різних розрізах. Отримані дані можуть оброблятися і бути видимими клієнтом різними способами залежно від потреб кінцевого користувача.

4. Можливість часткового оновлення даних. За допомогою XML можна оновлювати тільки ту частину структурованих даних, яка була змінена, а не всю структуру цілком.

Всі ці переваги роблять XML незамінним інструментом для розробки гнучких засобів пошуку інформації в базах даних, могутніх трирівневих веб-додатків, а також додатків, що підтримують транзакції. Іншими словами, за допомогою XML можна формувати запити до баз даних різних структур, що дозволяє здійснювати пошук інформації в численних несумісних один з одним базах даних. Використання XML на середньому рівні трирівневих веб-додатків дозволяє здійснювати ефективний обмін даними між клієнтами і серверами систем електронної комерції.

Крім того, мова XML може використовуватися як засіб для опису граматики інших мов і контролю правильності складання документів.

Інструменти обробки даних, отриманих у форматі XML, можуть бути розроблені в середовищі Visual Basic, Java або C++.

PHP - скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP — проект відкритого програмного забезпечення.

PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий html-код. Це є перевага з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніщо не забороняє використовувати PHP для генерування і JavaScript-кодів які виконуються вже на стороні клієнта.

PHP — мова, код якої можна вбудовувати безпосередньо в html-код сторінок, які, у свою чергу, будуть коректно оброблені PHP-інтерпретатором.

Обробник PHP просто починає виконувати код після відкриваючого тегу і продовжує виконання до того моменту, поки не зустрине закриваючий тег.

Велика різноманітність функцій PHP дає можливість уникати написання багаторядкових функцій, призначених для користувача, як це відбувається в C або Pascal.

В PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase.

Через стандарт відкритого інтерфейсу зв'язку з базами даних Open Database Connectivity Standard (ODBC) можна підключатися до всіх баз даних, до яких існує драйвер.

Мова PHP здаватиметься знайомою програмістам, що працюють в різних областях. Багато конструкцій мови запозичені з C, Perl. Код PHP дуже схожий на той, який зустрічається в типових програмах на C або Pascal. Це помітно знижує початкові зусилля при вивченні PHP. PHP — мова, що поєднує переваги Perl і C і спеціально спрямована на роботу в Інтернеті, мова з універсальним і зрозумілим синтаксисом. І хоча PHP є досить молодою мовою, вона здобула таку популярність серед web-програмістів, що в наш час є мало не найпопулярнішою мовою для створення веб-застосунків.

Стратегія Open Source, і розповсюдження початкових текстів програм в масах, безсумнівно справили благотворний вплив на багато проектів, в першу чергу — Linux хоч і успіх проекту Apache сильно підкріпив позиції прихильників Open Source. Сказане відноситься і до історії створення PHP, оскільки підтримка користувачів зі всього світу виявилася дуже важливим чинником в розвитку проекту PHP. Ухвалення стратегії Open Source і безплатне розповсюдження початкових текстів PHP надало неоціненну послугу користувачам. Окрім цього, користувачі PHP в усьому світі є свого роду колективною службою підтримки, і в популярних електронних конференціях можна знайти відповіді навіть на найскладніші питання.

Ефективність є дуже важливим чинником у програмуванні для середовищ розрахованих на багато користувачів, до яких належить і web. Важливою

перевагою PHP є те, що ця мова належить до інтерпретованих. Це дозволяє обробляти сценарії з достатньо високою швидкістю. За деякими оцінками, більшість PHP-сценаріїв (особливо не дуже великих розмірів) обробляються швидше за аналогічні їм програми, написані на Perl. Проте хоч би що робили розробники PHP, виконавчі файли, отримані за допомогою компіляції, працюватимуть значно швидше — в десятки, а іноді і в сотні разів. Але продуктивність PHP достатня для створення цілком серйозних веб-застосунків.

JavaScript (JS) — динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. Мова JavaScript також використовується для програмування на стороні серверу (подібно до таких мов програмування, як Java і C#), розробки ігор, стаціонарних та мобільних додатків, сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF-документів тощо. JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme.

JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від

традиційних мов ООП. Крім того, JavaScript має ряд властивостей, притаманних функціональним мовам, — функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання (closures) — що додає мові додаткову гнучкість.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою C 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

JavaScript містить декілька вбудованих об'єктів: Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. Крім того, JavaScript містить набір вбудованих операцій, які, строго кажучи, не обов'язково є функціями або методами, а також набір вбудованих операторів, що управляють логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови Java (тобто, зрештою, успадкований від C), але спрощений порівняно з ним, щоб зробити мову сценаріїв легкою для вивчення.

Так, приміром, декларація змінної не містить її типу, властивості також не мають типів, а декларація функції може стояти в тексті програми після неї. Семантика мови схожа з семантикою мови Self.

При розробці великих і нетривіальних веб-застосунків з використанням JavaScript, критично важливим є доступ до інструментів зневадження. Оскільки браузери від різних виробників дещо відрізняються у поведінці JavaScript і реалізації Об'єктної Моделі Документа, треба мати в руках зневаджувач для кожного браузера, якщо веб-застосування орієнтовано на нього.

Internet Explorer має три зневаджувача для себе: Microsoft Visual Studio є найпотужнішим з цих трьох, слідом йде Microsoft Script Editor (компонента Microsoft Office), і нарешті існує безкоштовний Microsoft Script Debugger з базовими функціями. Веб-застосування для Firefox допоможе привести до розуму додаток Firebug (зручно вбудований безпосередньо в браузер), або давніший відладчик Venkman, котрий також працює з браузером Mozilla. Drosera — це зневаджувач з WebKit engine, що супроводжує Apple Safari.

Також існують кілька інструментів, як вільних, наприклад JSLint, інструмент перевірки якості коду, що сканує JavaScript програму, шукаючи проблеми коду, так і комерційних продуктів типу інструменту з назвою JavaScript Debugger.

Оскільки JavaScript є інтерпретатором, без строгої типізації, і може виконуватися в різних середовищах, кожне зі своїми власними особливостями сумісності, програміст має бути дуже уважним, і повинен перевіряти, що його код виконується як очікується в широкому переліку можливих конфігурацій. Дуже часто трапляються випадки, коли скрипт, що чудово працює в одному середовищі, видає некоректні результати в іншому.

Кожен блок сценарію інтерпретатор розбирає окремо. На веб-сторінках, коли треба комбінувати блоки JavaScript та HTML, синтаксичні помилки знайти легше, якщо тримати функції сценарію в окремому блоці коду, або (ще краще) використовувати багато малих пов'язаних .js файлів. В такий спосіб синтаксична помилка не спричинятиме «падіння» цілої сторінки, і можна надати допомогу, елегантно вийшовши зі сторінки.

jQuery — популярна JavaScript-бібліотека з відкритим сирцевим кодом. Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків.

jQuery також надає можливості для розробників, для створення плагінів у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, розробники

можуть створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня. Це сприяє створенню потужних і динамічних веб-сторінок.

Основне завдання jQuery — це надавати розробнику легкий та гнучкий інструментарій кросбраузерної адресації DOM об'єктів за допомогою CSS та XPath селекторів. Також даний фреймворк надає інтерфейси для Ajax-застосунків, обробників подій і простої анімації.

Принцип роботи jQuery полягає в використанні класу (функції), який при звертанні до нього повертає сам себе. Таким чином, це дозволяє будувати послідовний ланцюг методів.

Для розробки використана система управління контентом Wordpress. Це зручний інструмент для керування контентом веб-сайту та розробкою під цю CMS. Система має зручний інтерфейс, невибаглива до ресурсів виділеного серверу.

3.4. Організаційне забезпечення

Під організаційним забезпеченням слід розуміти узгодження по місцю, часу і меті сумісне функціонування окремих виконавців, колективів і технічних засобів. Воно повинно здійснюватися і регулюватися деякими правилами взаємодії, які утворюють правовий та моральний кодекс і складають основу правового забезпечення. Тому організаційне забезпечення будується на нормативних актах правового забезпечення, а правове забезпечення знаходить своє втілення в організаційному забезпеченні.

Організаційне забезпечення інформаційної системи охоплює сукупність засобів, методів і відповідного персоналу. Воно повинно забезпечити:

- проведення техніко-економічного аналізу існуючої системи управління, вибору і постановки задач побудови інформаційної системи на етапі розробки і впровадження;

- регламентацію взаємодії персоналу з комплексом технічних засобів і між собою в процесі розв'язку задач управління, контролю ефективності роботи системи управління на етапі функціонування інформаційної системи.

На етапі проектування організаційне забезпечення виконує наступні задачі:

- аналіз існуючих систем управління і формулювання напрямів підвищення їх ефективності;
- вибір і постановку задач управління;
- формулювання вимог до комплексу технічних засобів;
- розробку організаційних рішень по складу, структурі, організації і методології розв'язку задач управління в інформаційній системі, склад робочих процедур і пояснення щодо їх виконання.

На етапі функціонування інформаційної системи організаційне забезпечення вирішує такі задачі:

- впровадження методів задач управління;
- організацію функціонування персоналу і комплексу технічних засобів інформаційної системи;
- контроль і аналіз ефективності управління;
- формування пропозицій по вдосконаленню і розвитку інформаційної системи.

В склад організаційного інформаційного забезпечення включаються схеми структури управління і списки штатних розкладів, уніфіковані форми документів, відомості про системи морального і матеріального стимулювання, посадові інструкції.

Підрозділами, які забезпечують безпосереднє функціонування системи керування турагентством є:

- відділ по роботі з клієнтами, який складає штат менеджерів, що займаються обслуговуванням нових заявок та вирішенням виникаючих проблем та виступають як звено між клієнтом та представником послуг.

- відділ програмного забезпечення, що займається підтриманням системи у належному стані та займається профілактичним обслуговуванням та вдосконаленням системи.

До того ж до користування і розробки системи причетне керівництво компанії, бухгалтерія та компанії-партнери (у разі користування).

ВИСНОВОК

Підчас роботи над проектом було вивчено та використано чимало засобів створення веб-сторінок, веб-додатків тощо. У результаті було розроблено багатофункціону систему управління взаємодії з клієнтами, що дає змогу автоматизувати ведення справ у туристичному агентстві та підвищує продуктивність підприємства та його привабливість для клієнтів.

Були використанні знання з таких мов програмувань, засобів розробки та фреймворків, як HTML5, JavaScript, JQuery, PHP5 (Laravel Framework), MySQL та CSS3. Проект був максимально відлагоджений, скрипти мінімізовані, база даних проіндексована та інші процедури, які сприяють підвищенню швидкості роботи ресурсу.

СПИСОК ЛІТЕРАТУРИ

1. Rahimi R. Impact of Customer Relationship Management on Customer Satisfaction: The Case of a Budget Hotel Chain / R. Rahimi, M. Kozak // Journal of Travel & Tourism Marketing, 2016.– №34 (1). – P. 40-51.
2. Mosa A. Customer Relationship Management: A Review and Classification / A. Mosa, A. Mais, A. Muneer, J. Faten // Transnational Marketing Journal, 2019. – № 2(7). – P. 187-210.
3. Юрчук Н. П. CRM-системи: особливості функціонування та аналіз українського ринку / Н. П. Юрчук // Науковий вісник Ужгородського національного університету, 2019. – № 23 (2). – С. 141-147.
4. Птащенко О. В. Побудова CRM-системи як основи формування комунікаційної політики між організацією та кінцевим споживачем / О.В. Птащенко, Є.Д. Мірошникова // Вісник Східноукраїнського національного університету імені Володимира Даля, 2016. – № 6 (230). – С. 108-115.
5. Томашевський О.М. Інформаційні технології та моделювання бізнес-процесів : Навчальний посібник / Томашевський О.М., Цегелик Г.Г., Вітер М.Б., Дубук В.І. // К.: Видавництво «Центр учбової літератури», 2012. - 296 с.
6. Tang C. The data industry: the business and economics of information and big data / C. Tang // Hoboken, New Jersey: John Wiley & Sons, 2016. – 195 p.
7. Tarla N. Microsoft Dynamics CRM 2016 Customization : 2nd Ed. / N. Tarla // – Packt Publishing, 2017. – 374 p.
8. Felke-Morris T. Web Development and Design Foundations with HTML5 : 7th edition / T. Felke-Morris // Pearson, 2015. – 696 p.
9. Caya A. Mastering the Faster Web with PHP, MySQL and JavaScript: Develop state of the art Web applications using the latest Web technologies / A. Caya // Packt Publishing, 2018. – 308 p.
10. Nixon R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 : 5th ed. / R. Nixon // Oreilly, 2018. – 832 p.

ДОДАТКИ

ДОДАТОК А

Лістинг основних частин коду

Звернення до контролеру:

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\Storage;

class Project extends Model
{
    protected $dates = ['created_at', 'updated_at', 'accepted_at'];

    public function client()
    {
        return $this->belongsTo(Client::class);
    }

    public function invoices()
    {
        return $this->hasMany(Invoice::class);
    }

    protected static function boot()
    {
        parent::boot();
        static::deleting(function ($project) {
            $project->invoices->each->delete();
            Storage::delete($project->pdf_path);
        });
    }
}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Invoice extends Model
{
    protected $fillable = ['item_details', 'due_date', 'paid', 'notes',
'paid_at'];
    protected $dates = ['created_at', 'updated_at', 'paid_at',
'due_date'];

    //    protected $with = ['client'];
```

```

public function client()
{
    return $this->belongsTo(Client::class);
}

public function project()
{
    return $this->belongsTo(Project::class);
}

protected static function boot()
{
    parent::boot();
    static::deleting(function ($invoice) {
        RecurringInvoice::where('invoice_id', $invoice->id)-
>delete();
    });
}
}

<?php

namespace App\Models;

use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use Notifiable;
    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password', 'role', 'is_admin', 'id',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    /**
     * Returns true if the user is an admin and
     * false if not.
     *
     */
}

```

```

        * @return bool
    */
    public function isAdmin()
    {
        return $this->is_admin;
    }

    public function isSuperAdmin()
    {
        if ($this->id === 1) {
            return true;
        } else {
            return false;
        }
    }

    public function client()
    {
        return $this->hasOne(Client::class);
    }
}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class RecurringInvoice extends Model
{
    protected $fillable = ['invoice_id', 'next_run', 'last_run',
'due_date'];
    protected $dates = ['created_at', 'updated_at', 'next_run',
'last_run'];

    public function invoice()
    {
        return $this->belongsTo(Invoice::class);
    }
}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Client extends Model
{
    protected $fillable = ['name', 'email', 'number', 'address',
'stripe_customer_id'];

    public function invoices()

```

```

    {
        return $this->hasMany(Invoice::class);
    }

    public function user()
    {
        return $this->belongsTo(User::class);
    }

    public function projects()
    {
        return $this->hasMany(Project::class);
    }

    public function recurringInvoices()
    {
        return $this->hasMany(RecurringInvoice::class);
    }

    public function addInvoice(Invoice $invoice)
    {
        return $this->invoices()->save($invoice);
    }

    public function addProject(Project $project)
    {
        return $this->projects()->save($project);
    }

    protected static function boot()
    {
        parent::boot();
        static::deleting(function ($client) {
            $client->invoices()->delete();
            $client->projects->each->delete();
            $client->recurringInvoices()->delete();
            $client->user->delete();
        });
    }
}

```

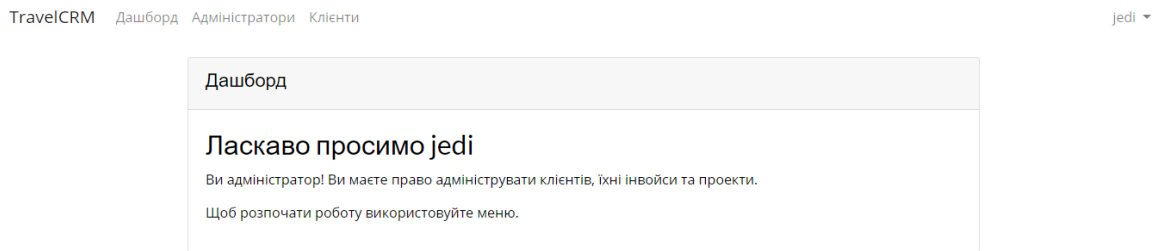


Рисунок Б.1 – Головна сторінка CRM

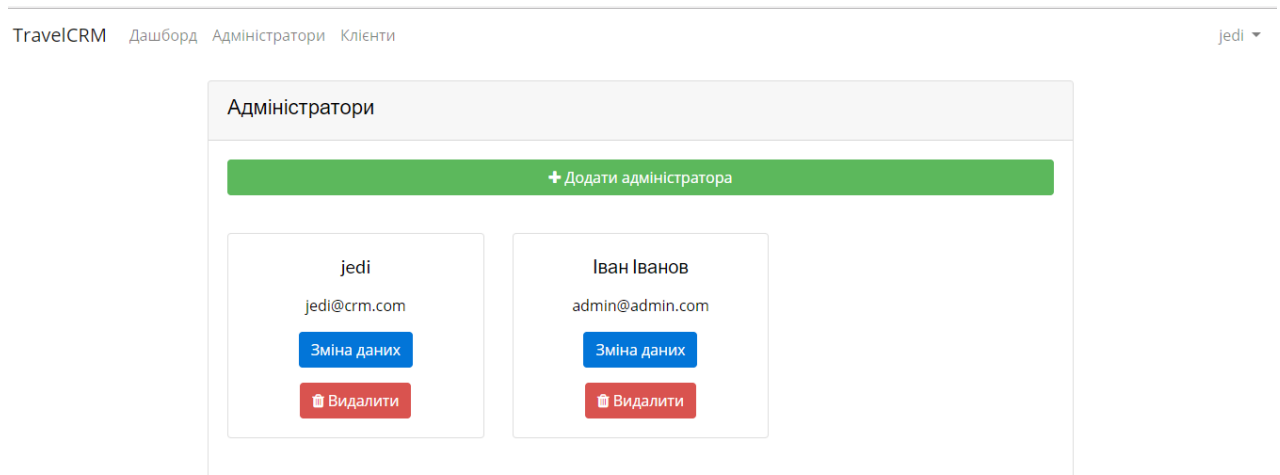


Рисунок Б.2 – Вікно керування доступом до системи

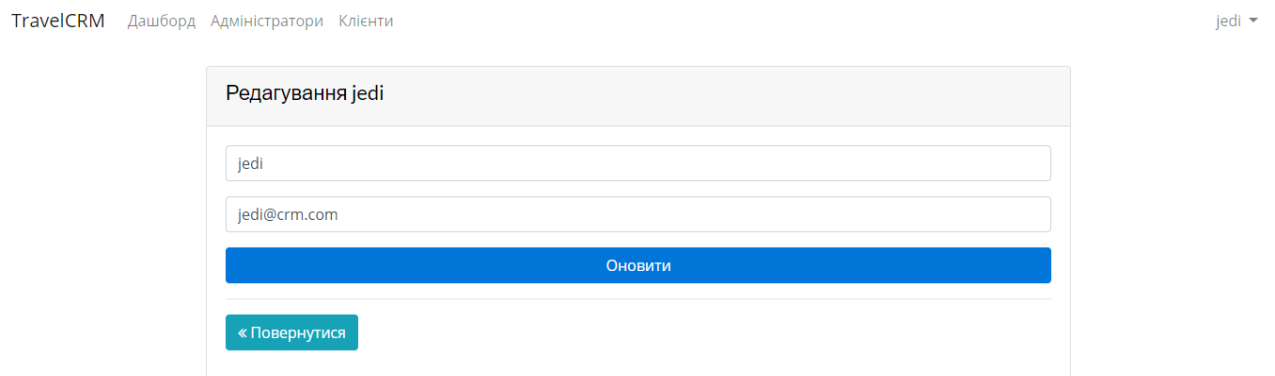


Рисунок Б.3 – Редагування головного адміністратора

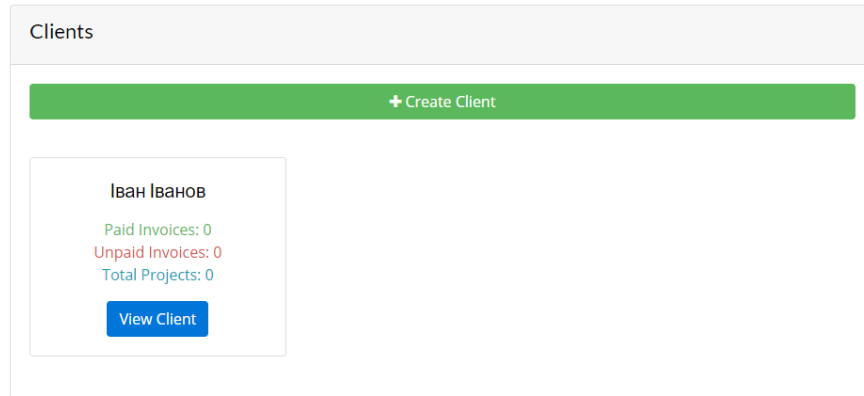


Рисунок Б.4 – Вікно перегляду клієнтів системи

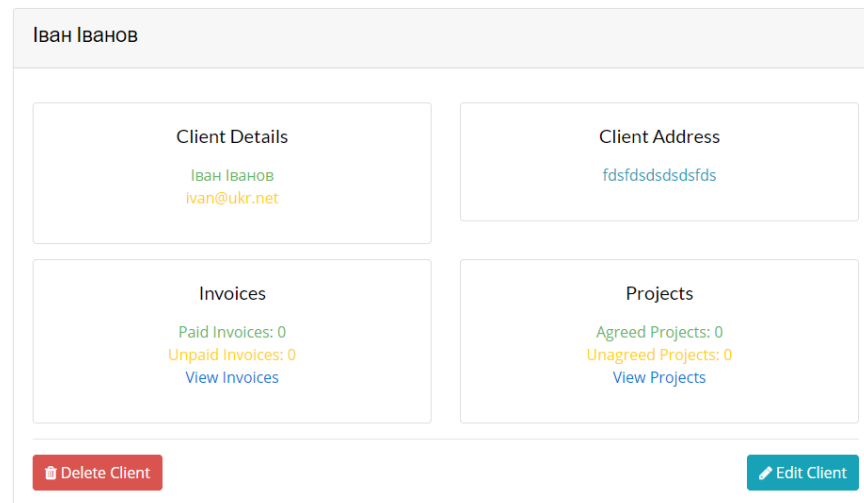


Рисунок Б.5 – Вікно перегляду стану заявок від клієнта

Edit Іван Іванов

Іван Іванов

ivan@ukr.net

Contact Number

fdsfsdsdsdsfsds

Update Client

← Back to Client

Рисунок Б.6 – Вікно керування клієнтом

1 Step 1 2 Step 2 3 Step 3

Fill in contact data.

BOOKING Draft **CLIENT** Contacts **VILLA** Choose

Booking draft

C-IN Date 01.06.2016 C-OUT Date

C-IN Time 14:00 C-OUT Time 10:00

Nights: 0

Available book villa(s)

Next Step **Draft**

Рисунок Б.7 – Заповнення даних броні (крок 1)

1 Step 1 2 Step 2 3 Step 3

Choose villa.

BOOKING Draft **CLIENT** Contacts **VILLA** Choose

Booking Data

First Name

Last Name

Middle Name

Phone Name

Email Name

[Previous Step](#) [Next Step](#)

Рисунок Б.8 – Заповнення даних броні (крок 2)

BOOKING Draft **CLIENT** Contacts **VILLA** Choose

Villa Data

C-IN Date C-OUT Date

C-IN Time : C-OUT Time :

Nights: 0

Available book villa(s)

Number of persons

Breakfast
Yes No

Number of breakfast

[Previous Step](#) [Send](#)

Рисунок Б.9 – Заповнення даних броні (крок 3)