

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ БРОНЮВАННЯ
МІСЦЬ В ЗАКЛАДАХ, ЩО НАДАЮТЬ ПОСЛУГИ ВІРТУАЛЬНОЇ
РЕАЛЬНОСТІ, МОВОЮ JAVASCRIPT»**

Виконав студент 5 курсу, групи ППЗ-52
спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Коваль О.С.

(прізвище та ініціали)

Керівник Гаманюк І.М.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

КИЇВ – 2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти Бакалавр
Спеціальність 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри ІІЗ
О.В.Негоденко
« » 2021 року

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Ковалю Олександр Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка мобільного додатку для бронювання місць в закладах, що надають послуги віртуальної реальності, мовою JavaScript»

Керівник роботи: Гаманюк Ігор Михайлович старший викладач кафедри

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «12» березня 2021 року
№ 65

2. Строк подання студентом роботи « 01» червня 2021 року.

3. Вихідні дані до роботи _____

1) Мережа інтернет. _____

2) Веб-сервер. _____

3) Технічна література. _____

4) Персональний комп'ютер

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз загальних відомостей про мобільні додатки.
2. Дослідження актуальності розробки мобільного додатку.
3. Аналіз вимог до програмного забезпечення.
4. Моделювання та проектування гібридного мобільного додатку.
5. Реалізація мобільного додатку.
6. Перелік графічного матеріалу (назва основних слайдів)
 1. Титульний слайд
 2. Мета, об'єкт та предмет дослідження
 3. Актуальність дослідження
 4. Статистика
 5. Вимоги
 6. Діаграма прецедентів
 7. Діаграма використання Адміністратор
 8. Діаграма використання Клієнт
 9. Схема побудови гібридного мобільного додатку
 10. Ключові елементи обраного підходу
 11. Діаграма розгортання
 12. Діаграма предметної галузі
 13. Діаграма артефактів (структура програми)
 14. Схема бази даних mysql
 15. Екран бронювання
 16. Діаграма пакетів екрану бронювання
 17. Екран керування замовленнями
 18. Діаграма пакетів екрану керування замовленнями
 19. Екран програми лояльності
 20. Екран статистики
 21. Висновки
6. Дата видачі завдання _____ « 19» квітня 2021 року _____.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1.	Аналіз загальних відомостей про мобільні додатки.	19.04.2021р.	виконано
2.	Дослідження актуальності розробки мобільного додатку.	22.04.2021р.	виконано
3.	Аналіз вимог до програмного забезпечення.	26.04.2021р.	виконано
4.	Моделювання та проектування гібридного мобільного додатку.	29.04.2021р.	виконано
5.	Реалізація мобільного додатку.	03.05.2021р.	виконано
6.	Реферат, вступ, висновки.	10.05.2021р.	виконано
7.	Підготовка презентації до захисту.	12.05.2021р.	виконано

Студент

(підпис)

Коваль О.С.

(прізвище та ініціали)

Керівник роботи

(підпис)

Гаманюк І.М.

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи: 88 с., 52 рис., 6 джерел.

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ БРОНЮВАННЯ МІСЦЬ В ЗАКЛАДАХ, ЩО НАДАЮТЬ ПОСЛУГИ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ, МОВОЮ JAVASCRIPT.

Об'єктом дослідження – є процес надання послуг користувачам.

Предмет дослідження – автоматизація процесу бронювання місць шляхом створення відповідного програмного забезпечення.

Мета роботи – покращення методики надання послуг користувачам шляхом використання розробленого програмного забезпечення.

Методи дослідження .У науковій роботі було використані загальнонаукові та спеціальні методи дослідження, а саме: методи аналізу та синтезу, індукції та порівняння під час аналізу вимог та моделюванню програмного забезпечення, також графічні методи – для представлення результатів дослідження у вигляді схем та рисунків тощо.

Визначено методи та засоби розробки гібридних мобільних додатків.

Здійснено аналіз предметної галузі та технологій які використовують для забезпечення роботи мобільного додатку, моніторинг ринку програмного забезпечення, а також розробка програмного забезпечення.

Результати дослідження можуть бути використані в галузі управління проектами з розробки гібридного ПЗ для мобільних телефонів

Упровадження розробленої методики дозволяє розробити мобільний додаток з використанням веб-технологій який буде працювати на Android та IOS в короткі терміни з дотриманням всіх вимог до програмного забезпечення.

ЗМІСТ

ВСТУП.....	10
1 АНАЛІЗ НАЯВНИХ ЗАСОБІВ ТА ТЕХНОЛОГІЙ ДЛЯ ОРГАНІЗАЦІЇ ПРОЕКТУ З РОЗРОБКИ ГІБРИДНОГО ПЗ ДЛЯ МОБІЛЬНИХ ТЕЛЕФОНІВ ...	7
1.1 Постановка проблеми.....	7
1.2 Ситуація використання додатків на ринку в Україні	8
1.3 Використання інтернету в Україні – дослідження	8
1.4 Про популярність мобільних платформ.....	15
1.5 Розробка додатків для android	15
2 ВИМОГИ (АНАЛІЗ ВИМОГ).....	18
2.1 Бізнес-вимоги	18
2.2 Вимоги користувача	18
2.3 Функціональні вимоги	19
2.4 Нефункціональні вимоги	23
3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ГІБРИДНОГО МОБІЛЬНОГО ДОДАТКУ	24
3.1 Інструменти моделювання	24
3.2 Розробка макета програми	26
4 СЕРЕДОВИЩЕ РОЗРОБКИ.....	27
4.1 Особливості Visual Studio Code	28
4.2 Можливості Visual Studio Code	28
4.3 Переваги Visual Studio Code	29
4.4 Емулятор Genymotion.....	29
4.5 Особливості Genymotion.....	31
4.6 Недоліки емулятора Genymotion	32
5 ТЕХНОЛОГІЇ ЯКІ Я ВИКОРИСТОВУВАВ.....	33
5.1 Фреймворк інтерфейсу Ionic Framework.....	33
5.2 Мова програмування Javascript	41
5.3 Бібліотека jQuery	46

5.4 Мова програмування PHP	50
5.5 Використання JSON серіалізації.....	55
5.6 Використання MySQL Бази даних	61
5.7 Побудова арк мобільного додатку за допомогою Cordova	62
6 РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ	63
6.1 Екран авторизації	63
6.2 Екран реєстрації користувачів.....	65
6.3 Екран відновлення пароля	67
6.4 Головний екран.....	69
6.5 Екран акцій	72
6.6 Екран онлайн бронювання.....	73
6.7 Екран програми лояльності	75
6.8 Екран контактів	77
6.9 Екран правил.....	79
6.10 Екран керування замовленнями.....	80
6.11 Екран статистики.....	83
6.12 Діаграма предметної галузі.....	84
6.13 Діаграма артефактів	85
7 РОЗГОРТАННЯ	86
ВИСНОВКИ	87
Висновок до пункту 5.1.....	87
Висновок до пункту 5.3.....	87
ПЕРЕЛІК ПОСИЛАНЬ	89
ДОДАТОК	90

ВСТУП

Інформаційні технології допомагають оптимізувати роботу, скоротивши витрати на власну діяльність. На практиці більшість компаній досягає цього та все більше автоматизує свою роботу. Новітні електронні технології впроваджуються в усіх сферах життя суспільства та організацій. Програмне забезпечення посідає важливе місце в управлінні, організації роботи, обслуговуванню клієнтів а інформація яка міститься в додатках, становить цінність. Якщо ще пару століть назад в різних частинах світу людей масово охоплювала Золота лихоманка, то зараз з упевненістю можна сказати, що сучасний світ цілком і повністю поглинений мобільною лихоманкою. Чи знайдеться у вашому оточенні хоча б одна людина, яка не користується смартфоном? Продажі смартфонів зростають з блискавичною швидкістю, а компанії-виробники вкладають величезні суми в мобільні технології, їх розвиток і популяризацію на ринку.

Мобільна розробка – дуже стрімко зростаюча область програмування, адже кількість мобільних пристроїв значно перевищує кількість персональних комп'ютерів, і ця тенденція буде тільки зростати. Ми продовжуємо користуватися мобільними телефонами всюди – на роботі, на навчанні, вдома, на відпочинку ... Такими ж темпами зростає і попит на розробників мобільних додатків, але на відміну від фронтенд і backend девелоперів, ринок ще не насичений мобільними розробниками, тому зараз – саме час починати освоювати це перспективний напрямок. Тому розробка мобільного додатку досить актуальна тема.

Актуальність дослідження. Смартфони та інші мобільні пристрої не тільки стали частиною нашого повсякденного життя, вони – повноцінне продовження нас. За допомогою мобільних телефонів ми не тільки спілкуємося один з одним, але і замовляємо товари з магазинів, купуємо квитки, бронюємо житло, викликаємо таксі, використовуємо телефони як навігатори, фото- і відеокамери, читалки, онлайн-банки, і просто як спосіб розважитися і скоротати час. Згідно зі

статистикою, опублікованою в Datareportal, 67% дорослих людей в усьому світі використовують смартфони щодня, а це майже 5,19 млрд осіб (при загальній кількості населення в 7,75 млрд). Тенденція до переходу з простих мобільних пристроїв на багатофункціональні смартфони з кожним роком тільки збільшується. Передбачається, що в найближчому майбутньому людина буде безпосередньо пов'язана зі своїм смартфоном, тобто всі функції і вся інформація з наших гаджетів буде прямо надходити в наш мозок. Коротше кажучи, актуальність розробки мобільних додатків зростає не те що з кожним роком, а й з кожним місяцем. Сотні нових мобільних додатків виходять на онлайн-майданчиках кожен день. Здавалося б, складно придумати щось нове, але хороші розробки в колаборації з хорошим маркетингом можуть принести багато грошей творцеві, явне тому підтвердження – гучний старт TikTok, рейтинги якого продовжують блискавично рости.

Метою дослідження є покращення методики надання послуг користувачам шляхом використання розробленого програмного забезпечення.

Об'єктом дослідження є процес надання послуг користувачам.

Предметом дослідження є автоматизація процесу бронювання місць шляхом створення відповідного програмного забезпечення.

Результати дослідження можуть бути використані в галузі управління проектами з розробки гібридного ПЗ для мобільних телефонів.

Завдання дослідження. В процесі дослідження вирішувалися наступні завдання:

- Аналіз вимог,
- Проектування,
- Реалізація,
- Тестування і налагодження,
- Впровадження,
- Супровід.

Як й інші традиційні інженерні дисципліни, розробка програмного забезпечення має справу з проблемами якості, вартості та надійності. Деякі програми містять мільйони рядків вихідного коду, які, як очікується, повинні правильно виконуватися в умовах, що змінюються. Складність ПЗ порівнянна зі складністю найбільш складних з сучасних машин, таких як літаки.

Наукова новизна дослідження полягає в моделюванні процесу бронювання місць клієнтами в закладах, що надають послуги віртуальної реальності а також керування і обробкою цих даних.

Практична значущість результатів дослідження полягає в отриманні моделі процесу бронювання місць клієнтами в закладах, що надають послуги віртуальної реальності а також керування і обробкою цих даних, як складової моделі загального процесу бронювання і обробки даних з використанням інформаційних технологій.

1 АНАЛІЗ НАЯВНИХ ЗАСОБІВ ТА ТЕХНОЛОГІЙ ДЛЯ ОРГАНІЗАЦІЇ ПРОЕКТУ З РОЗРОБКИ ГІБРИДНОГО ПЗ ДЛЯ МОБІЛЬНИХ ТЕЛЕФОНІВ

1.1 Постановка проблеми

У процесі керування та підтримки бізнесу інформаційні системи, що використовують комп'ютерні технології, відіграють дуже важливу роль, оскільки оптимізація роботи та контроль за її виконанням, важливими показниками підвищує успіхи в отриманні підвищення прибутку та надходженню нових клієнтів в результаті покращення обслуговування. В якості одного з перспективних методів сучасні інформаційні технології пропонують автоматизацію яка моделюється комп'ютером і розглядається в якості особливого інформаційного середовища.

Світовий розвиток інформаційних технологій можна розділити на 4 фази: предмеханічний (3000 до н.е.-1450 н.е.), механічний (1450-1840 рр.), електромеханічний (1840-1940 рр.) та електронний (1940-теперішній час). Основною відмінністю четвертої фази розвитку людства є автоматизація процесів. Завдяки застосуванню різноманітних програмних продуктів людина сьогодні спроможна обробляти інформацію у 2400 разів більше, ніж навіть два десятиліття тому.

Слідом за і-технологіями трансформується і бізнес з основним гаслом "Все для клієнта, все заради клієнта". Але виграти боротьбу за клієнта без застосування потрібного програмного забезпечення у сучасних умовах просто неможливо, адже сучасний темп життя вимагає від нас бути більш мобільними, прогресивними та ефективними.

В онлайн-просторі бізнес масштабується, географія аудиторії магазинів розширюється, а разом зі стрімким розвитком e-commerce збільшується і швидкість укладання угод. Наприклад, за прогнозами дослідницької компанії

eMarketer, до 2021 року частка мобільної торгівлі на світовому ринку електронної комерції складе 72,9% від усіх продажів. Статистика найбільших маркетплейсів, де сьогодні зосереджена велика кількість бізнесів, відображає дану ситуацію найкраще. Так, український e-commerce майданчик OLX у 2018 році мінімум раз на місяць відвідує 50% української інтернет-аудиторії. 60% використовують для цього мобільні пристрої. Такі дані навели у прес-службі OLX.

1.2 Ситуація використання додатків на ринку в Україні

Цей тренд підтверджують і результати дослідження GfK. Ще в 2017 році в Україні було тільки 82% користувачів смартфонів. Зараз їх кількість досягла 85%. При цьому за статистикою кожен споживач у середньому має 3-4 гаджети. Вивчення процесу вибору і покупки товарів українцями показало, що 50% людей шукали інформацію про товар зі смартфонів, і внаслідок, 43% зробили замовлення за допомогою мобільного телефону.

Понад 50% покупок здійснюється українцями через спеціально встановлені додатки. За даними GfK, найпопулярнішими серед них вже котрий рік залишаються OLX та AliExpress. Аналітика додатку OLX свідчить про те, що за I півріччя 2018-го кількість його користувачів зросла на 14% і склала 5,6 млн осіб.

1.3 Використання інтернету в Україні – дослідження

Майже 23 млн. українців регулярно користуються інтернетом.

Мобільний телефон або смартфон для доступу до мережі використовують 66% українців.

Інтернетом регулярно користуються 71% українців (22,96 млн). Про це йдеться у дослідженні Дослідницького холдингу Factum Group Ukraine.

За результатами дослідження, у 21 млн осіб є Інтернет вдома (65%).

Інтернет: основні показники



Рисунок 1.3.1 Регулярні користувачі та наявність інтернету вдома

Згідно з динамікою проникнення Інтернету, у III кварталі 2019 року 71% використовують Інтернет один раз на місяць або частіше, це на 7 в.п. більше ніж у I кварталі цього року (64%).

Динаміка проникнення Інтернету: щоквартальний замір

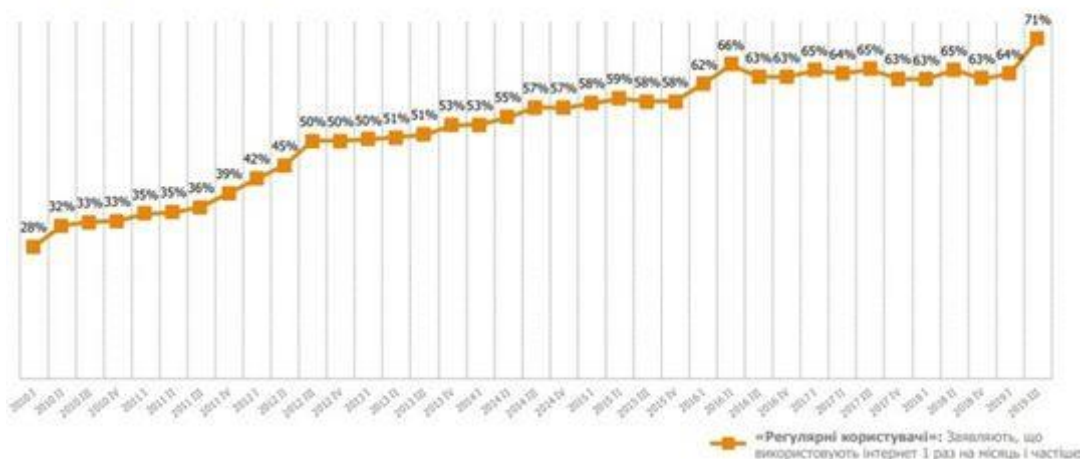


Рисунок 1.3.2 Динаміка користування інтернетом

У містах з населення більше 100 000 кількість користувачів Інтернету зросла на 3 в.п. (74%), ніж у 2018 році (71%). Вони складають 42% від всіх користувачів Інтернету в Україні.

Міста з населенням менше 100 000 регулярно використовують мережу на 7 в.п (70%) більше, ніж минулоріч (63%). Вони складають 29% від загальної кількості користувачів, також 29% складають жителі села.

Зростання використання Інтернету простежується й в селах, так у 2019 році показник зріс на 5 в.п. (58%), у 2018 році показник становив 53%.

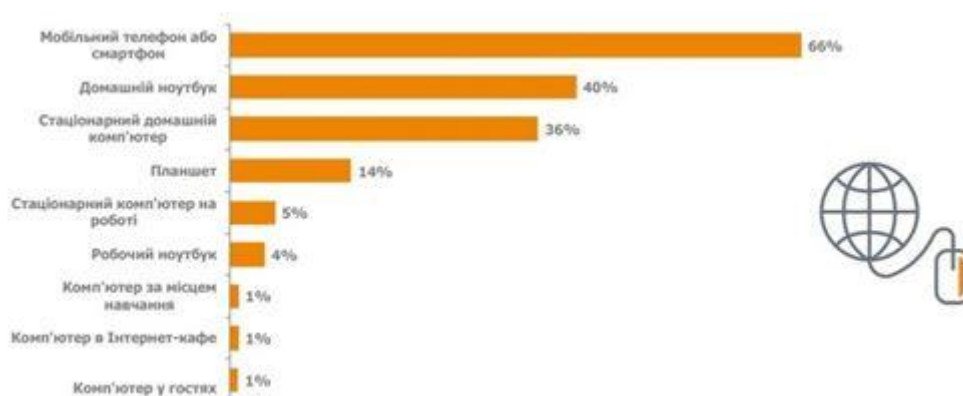


Рисунок 1.3.3 Щорічна динаміка

Дослідження показує, що найбільше Інтернетом користуються жінки 52%, чоловіки – 47%.

Найбільше користувачів мережі у Центрально-Північних регіонах України (33%), на Сході – 29%, Західні регіони – 27%. Південь країни досяг показника у 11%.

Найбільша кількість користувачів серед осіб віком від 25-34 років (25%). Друге місце по користуванню займають користувачі віком від 35 до 44 років – 21%. 18% – особи віком від 45 до 54, підлітки та молодь віком від 15 до 24 років складають лише 15%. 12% – віком від 55 до 64 років та 8% особи старші 65 років.

Соціально-демографічна структура «регулярних» інтернет-користувачів



Рисунок 1.3.4 Демографічна структура інтернет-користувачів

Серед користувачів 44% мають середній дохід, 40% – нижче середнього та 7% – мають низький рівень доходу. 6% користувачів мають дохід вище середнього і лише 1% мають високий рівень доходу.

Повну вищу освіту мають 36%, неповну вищу – 5%, середню спеціальну – 40%, повну середню мають – 16% та 3% мають неповну середню освіту.

Соціально-демографічна структура «регулярних» інтернет-користувачів

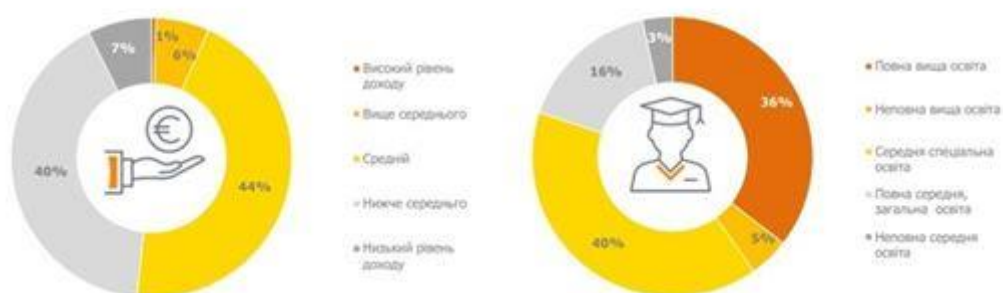


Рисунок 1.3.5 Демографічна структура інтернет-користувачів по освіті

Серед «регулярних» користувачів 21% – робітники, 16% – спеціалісти з вищою освітою. 60% одружені, а у 47% є діти до 16 років.

Соціально-демографічна структура «регулярних» інтернет-користувачів

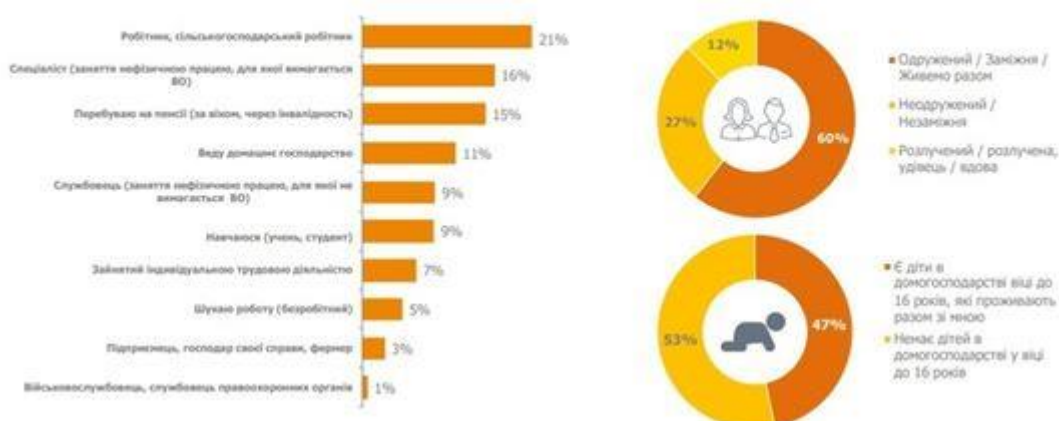


Рисунок 1.3.6 Соціально-демографічна структура інтернет-користувачів

У містах з населенням вище 100 000 75% «регулярних користувачів», менше 100 000 – 75% користувачів, а у селах – 62%. Серед осіб віком 15-24 роки 97% «регулярно» користуються Інтернетом, 25-34 роки – 96 %, 35-44 роки – 91%, 45-54 роки – 74%, 55-64 роки – 54% та 29% «регулярних» користувачів віком старше 65 років.

Доля «регулярних» інтернет-користувачів в розрізах

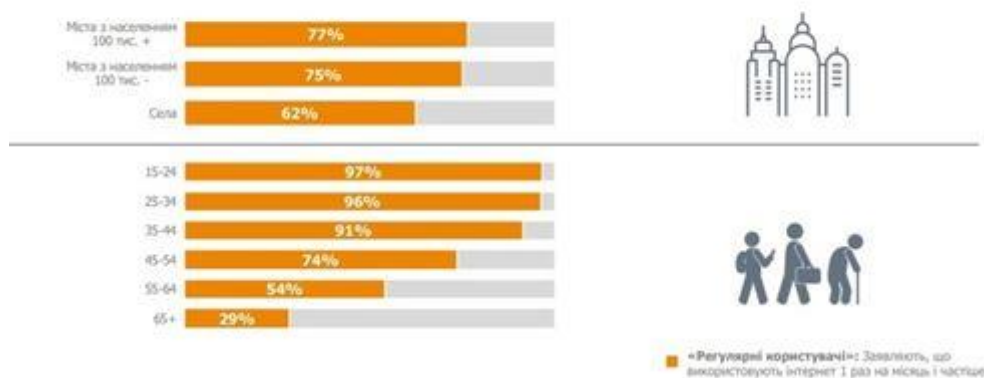


Рисунок 1.3.7 Статистика інтернет-користувачів в розрізі

Серед осіб з високим рівнем доходу 100% «регулярних користувачів», вище середнього – 93%, середній – 88%, нижче середнього – 66%, низький – 39%.

87% «регулярних» користувачів мають повну вищу освіту, 84% – неповну вищу, 72% – середня спеціальна, 54% – початкова, а повну середню мають 49%.

Доля «регулярних» інтернет-користувачів в розрізах

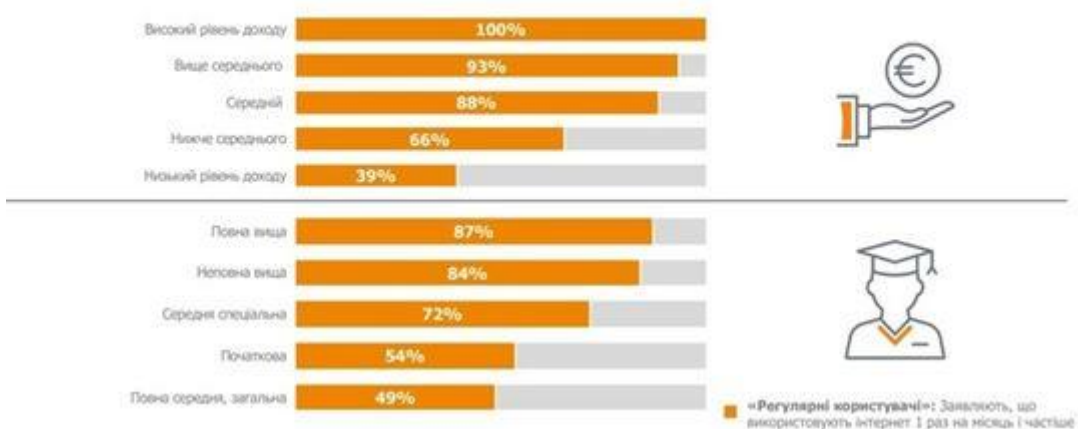


Рисунок 1.3.8 Статистика інтернет-користувачів за освітою та доходом

Найбільше «регулярних» користувачів серед учнів та студентів (97%). Неодружених та незаміжніх «регулярних» користувачів 86%.

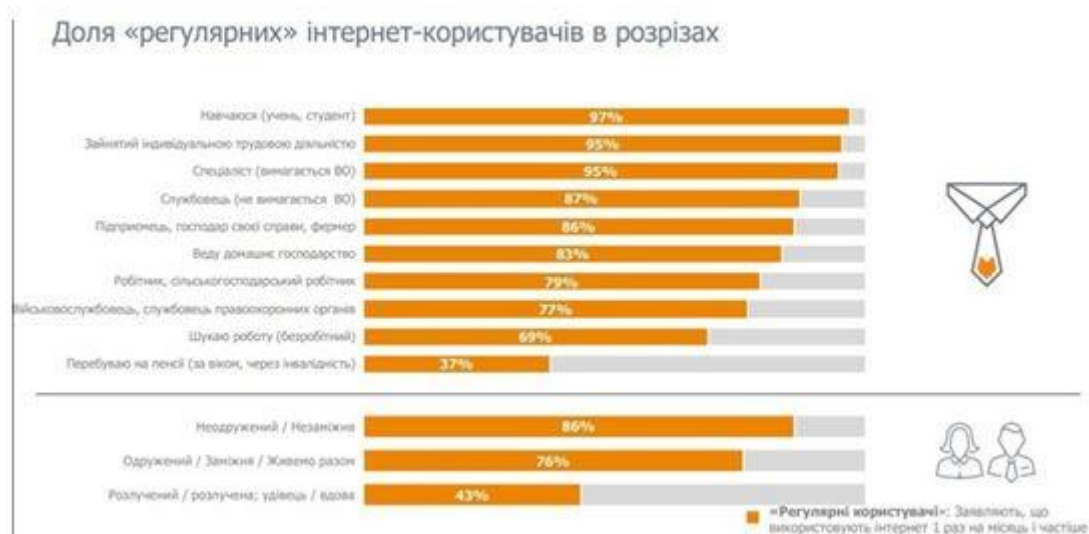


Рисунок 1.3.9 Статистика інтернет-користувачів за критеріями

Мобільний телефон або смартфон використовують 66% користувачів, домашній ноутбук – 40%, стаціонарний домашній комп'ютер використовують 36%, а планшет лише 14%.

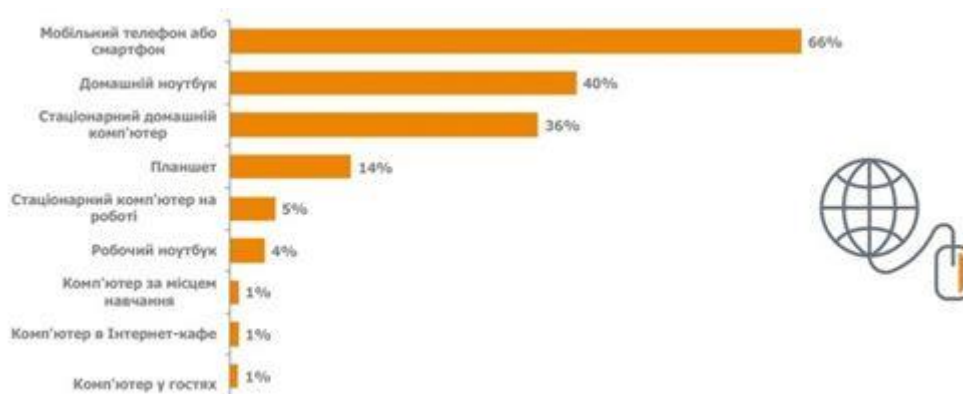


Рисунок 1.3.10 Статистика інтернет-користувачів за типом доступу

Серед «регулярних» користувачів 69% використовують мобільні пристрої (смартфон + планшет), 22% використовують лише мобільні пристрої.

51% користувачів мають тільки один пристрій для користування Інтернетом, 35% – два пристрої і 14% – ти або більше.

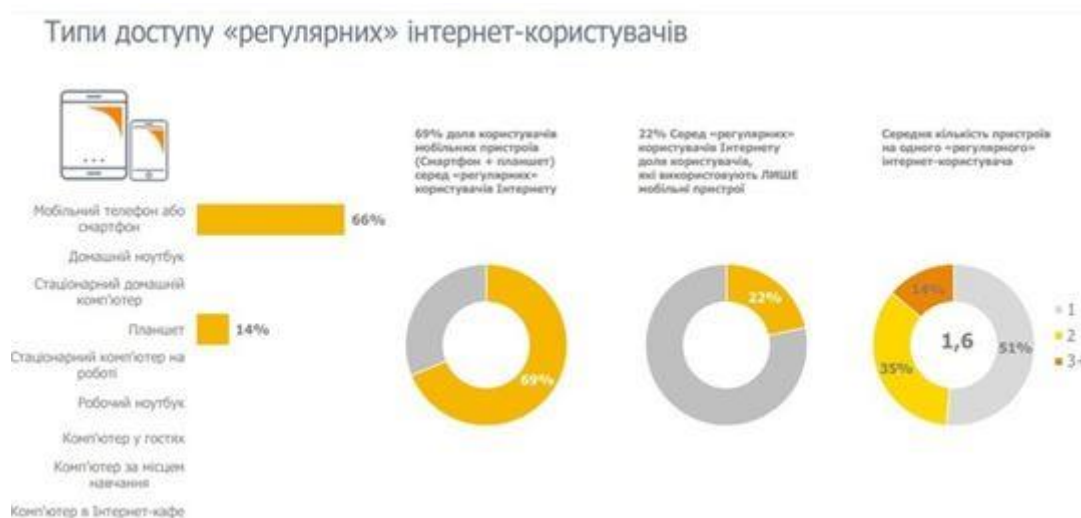


Рисунок 1.3.11 Статистика інтернет-користувачів за типом доступу

Дослідження було проведено у серпні – вересні 2019 року, шляхом face-to-face інтерв'ю за місцем проживання респондента. Для опитування використовувалися технічні засоби – планшети, репрезентативна населенню України, віком 15 років і старше. З дослідження виключена АР Крим та окупована територія Луганської і Донецької областей. Розмір генеральної сукупності 32,13 млн чоловік.

1.4 Про популярність мобільних платформ

Серед користувачів OLX 4,4 млн використовують Android. За 6 місяців ця цифра збільшилася на 18%. Інші 1,2 млн використовують пристрої на базі iOS. Темпи зростання завантажень серед власників iPhone у свою чергу склали 3%. Однак, загалом, спостерігається стабільна динаміка, супутня зростанню всього ринку, зазначили в компанії.

1.5 Розробка додатків для android

Сьогодні Android, мабуть, найпопулярніша в світі платформа для мобільних пристроїв. Android навколо нас – вдома, в офісі, в камерах і годинниках, в автомобілях і телевізорах. Додатки Android мають доступ до провідного відкритого торгового майданчика для поширення: Google Play. У магазині Google Play більше 1 мільйона додатків доступних для завантаження на ваш пристрій Android

1.5.1 Навіщо взагалі потрібні мобільні додатки, які типи мобільних додатків існують і як відбувається їх поширення.

Згідно зі статистикою BuildFire, компанії-розробники мобільних телефонів, як очікується, створять додатки, які до 2022 року завантажуть 258,2 мільйона завантажень проти 205,4 в 2018 році. Це означає, що дохід, що генерується компаніями, що займаються мобільними додатками, також швидко зростає. Причина цього полягає в тому, що смартфони пов'язані саме з мобільним програмним забезпеченням. Наприклад, Instagram або Uber – це мобільні додатки, які можна використовувати тільки на своєму смартфоні, тому Instagram має обмежену функціональність, якщо ви використовуєте його на комп'ютері, а

додаток Uber було розроблено для обслуговування користувачів, які знаходяться в дорозі.

1.5.2 Типи мобільних додатків

Мобільні додатки розрізняються залежно від операційної системи, в якій вони працюють – основними з яких є Android і iOS. Це означає, що, якщо ви хочете розробити програму, яка буде працювати для обох, вам доведеться використовувати різні комплекти розробки програмного забезпечення. Платформа iOS – це операційна система, яка підтримує популярну лінійку смартфонів iPhone від Apple. Другий – це Android від Google.

Три основних типи мобільних додатків, які можна розробити для iOS і Android:

- Рідні додатки. Додаток, призначений для конкретної операційної системи, називається власним. Засновані на власному інтерфейсі, такі додатки демонструють кращу продуктивність з пристроями, для яких вони призначені.

- Гібридні додатки. Додаток Facebook було створено на основі React Native – середовища розробки, заснованого на JavaScript, яка дозволяє створювати додатки з використанням бібліотеки блоків, які можуть обслуговуватися як для iOS, так і для Android. Це хороший приклад того, що являють собою гібридні програми – в основному, комбінація нативних і веб-елементів. Гібридний підхід стане в нагоді, якщо у вас обмежені ресурси і ви зосереджені на розробці кроссплатформного додатку. З іншого боку, гібридні програми демонструють більш низьку продуктивність у порівнянні з нативними і виявляються більш уразливими з точки зору безпеки.

- Веб-додатки. Додатки, в основному написані на HTML, CSS і JavaScript, які працюють у вашому браузері, + екранів, такі додатки можуть здатися трохи чужими для смартфона, особливо якщо вам необхідно представляти великий обсяг даних. Тому рекомендується використовувати цей підхід з додатками, які не зовсім легкі і займають занадто багато місця на мобільних пристроях, якщо розробляються спочатку.

1.5.3 Деякі ключові проблеми розробки додатків для Android

Android є найпопулярнішою ОС (англ. Operating system) для смартфонів у всьому світі і надає власнику додатку доступ до величезної бази користувачів. Однак розробники, незважаючи на безліч можливостей, стикаються з багатьма проблемами, про які важливо знати.

Фрагментація програмного забезпечення. Існує безліч версій ОС Android, з якими розробникам важко встигати за розробкою додатків. Недоцільно зосереджуватися тільки на самій останній версії Android, оскільки не всі користувачі могли перейти на останню версію ОС.

Фрагментація апаратного забезпечення: це стає серйозною проблемою для розробки додатків для Android, оскільки в операційній системі знаходиться більше 170 пристроїв. Кожен пристрій має різні функції щодо форм клавіатури, розміру екрана, кнопок камери тощо.

Відсутність стандартизації програмного / апаратного забезпечення: величезна кількість пристроїв під управлінням Android породжує ще одну проблему розробки додатків для Android – відсутність стандартизації програмного / апаратного забезпечення на всіх пристроях. Це стає кошмаром для розробників, оскільки кожен пристрій має різні функції для різних кнопок.

Безпека. На відміну від строгих правил Apple по розробці додатків, для додатків Android такого управління не існує. В результаті виникає багато проблем з шкідливим ПЗ, а фрагментація програмного і апаратного забезпечення тільки ускладнює вирішення проблем. Це викликає величезну кількість проблем безпеки.

Витрати на дослідження ринку. Однією з найсерйозніших проблем для розробників Android-додатків є вартість дослідження ринку. Розуміння кінцевого користувача є ключем до розробки додатків для Android, але може зажадати багато досліджень, що робить його дорогим для розробників.

2 ВИМОГИ (АНАЛІЗ ВИМОГ)

При аналізі вимог я користувався таким методом знаходження вимог як спілкування з майбутнім користувачем.

2.1 Бізнес-вимоги

За допомогою такого мобільного додатку організація ставить цілі покращити обслуговування клієнтів надаючи інформацію в зручному для користувача варіанті, а також підвищити потік клієнтів надаючи можливість накопичення кількості відвідувань і заохочувати постійних клієнтів бонусами, наприклад у вигляді безкоштовного відвідування закладу.

З боку працівників цей додаток має покращити мобільність в наданні послуг адже потреба в комп'ютері та перебуванні біля нього менеджера буде відсутня. Саме такі вимоги має намір досягти організація.

2.2 Вимоги користувача

Адміністратори:

- Можливість прибирати замовлення
- Можливість змінювати замовлення
- Показ нових замовлень
- Показ статистики замовлень за місяць

Користувачі:

- Можливість забронювати місце
- Показ основної інформації про заклад:
 - 1) Геолокація
 - 2) Контакти
 - 3) Нові ігри

- 4) Акції
- 5) Ціни на послуги
- Програма лояльності (знижки за кількістю відвідувань)

2.3 Функціональні вимоги

Функціональні вимоги представлені нижче в вигляді UML діаграм прецедентів та діаграм використання клієнтом та адміністратором.

Діаграма прецедентів — в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (англ. use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації (англ. association relationship)
- включення (англ. include relationship)
- розширення (англ. extend relationship)
- узагальнення (англ. generalization relationship)

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме — за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації — одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (англ. *include*) у мові UML — це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (англ. *dependency*) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (англ. *extend*) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

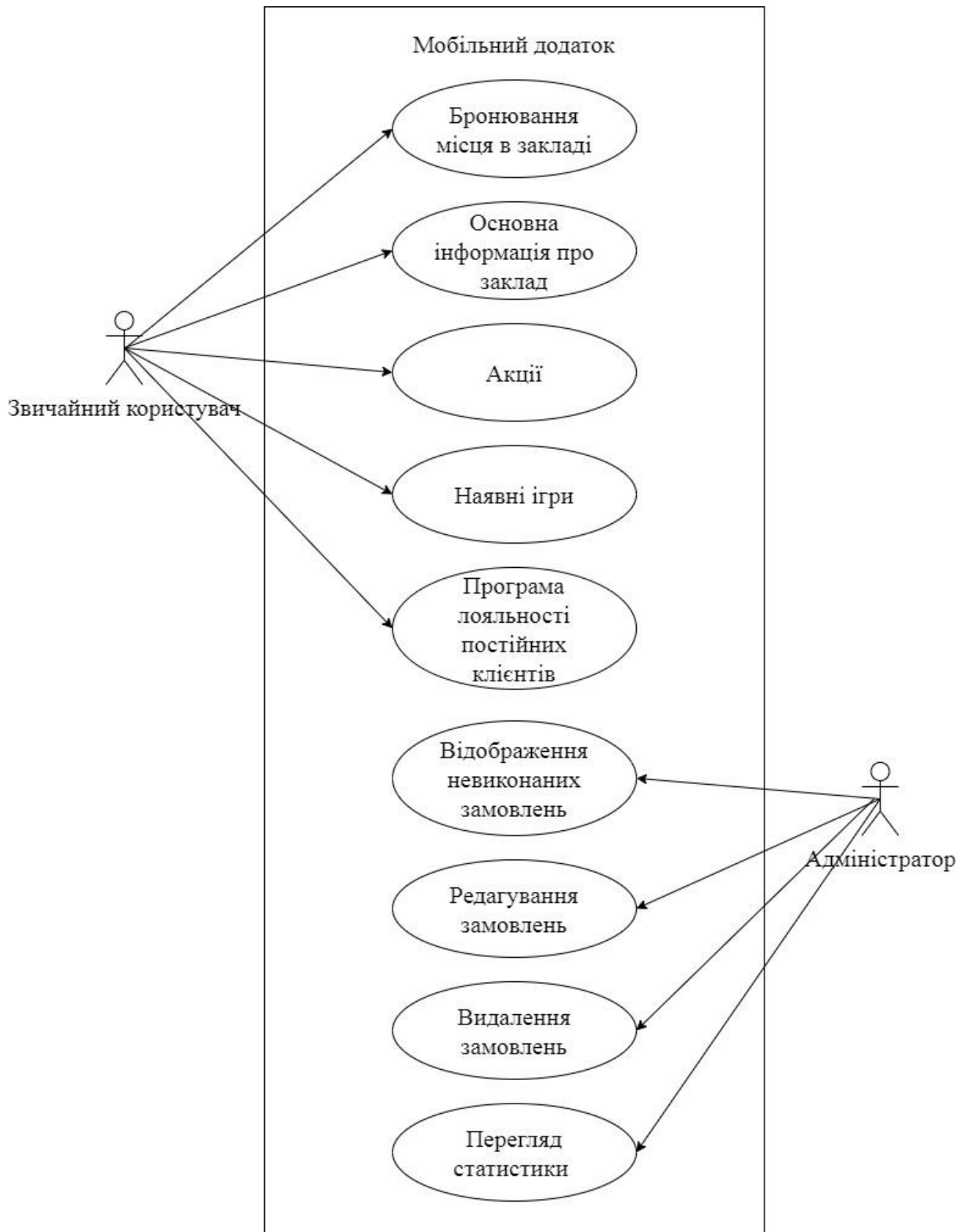


Рисунок 2.1 Діаграма прецедентів

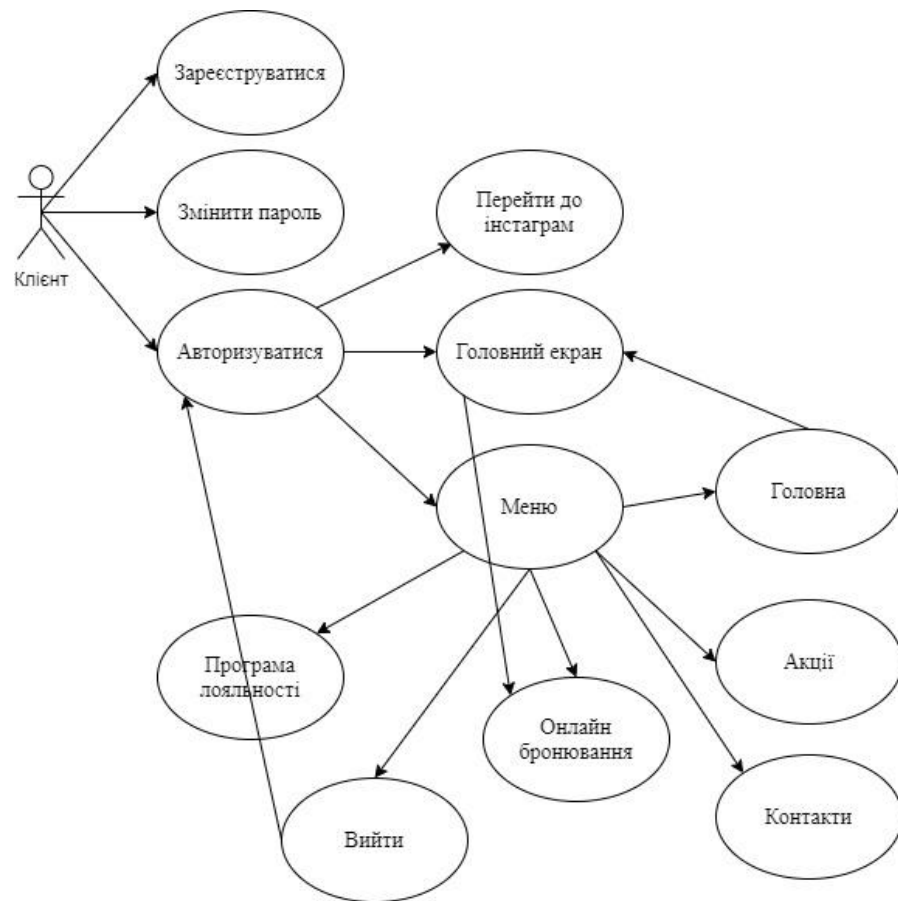


Рисунок 2.2 Діаграма варіантів використання клієнтом

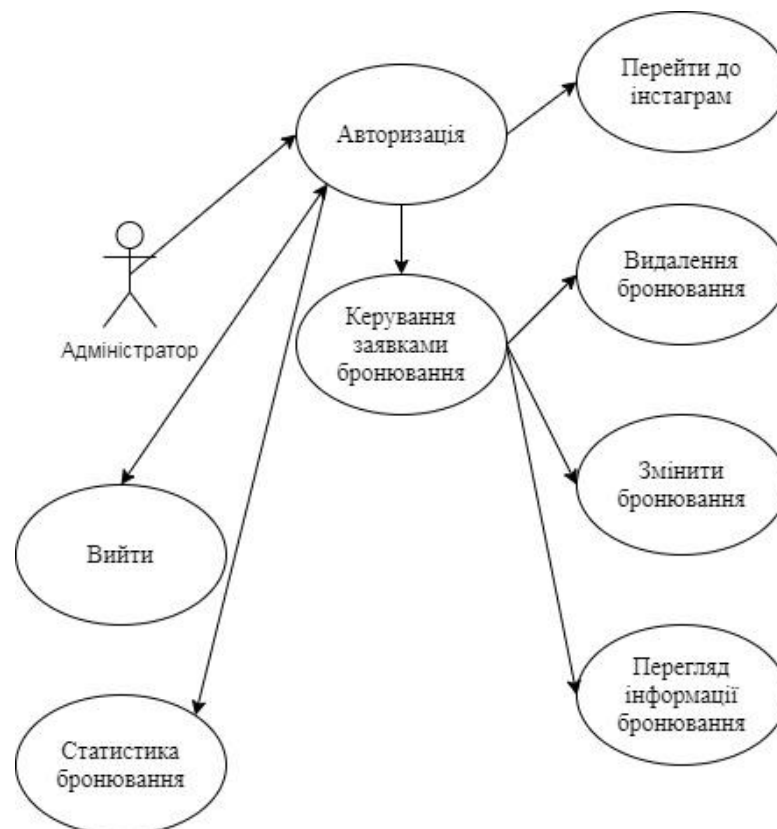


Рисунок 2.3 Діаграма варіантів використання Адміністратор

2.4 Нефункціональні вимоги

2.4.1 Вимоги до інтерфейсу

Апаратні інтерфейси :

- Модуль GSM або WIFI

Інтерфейси ПЗ :

- http,https запити

2.4.2 Апаратні та програмні вимоги

- Наявність доступу до мережі інтернет.
- Cordova підтримує Android 4.0.x (починаючи з рівня Android API 14) і вище. Як правило версії Android перестають підтримуватися Cordova після того як вони опустяться нижче 5% на панель поширення Google. Версії Android менш ніж API рівня 10, і версії 3.x (Honeycomb, рівні API 11-13) впали значно нижче цього 5% -го порогу.
- Налаштований сервер з базою даних mysql та php 5.4 і вище.
- Сервер повинен мати ssl сертифікат безпеки.

3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ГІБРИДНОГО МОБІЛЬНОГО ДОДАТКУ

3.1 Інструменти моделювання

При створенні моделі мобільного додатку я використовував такий інструмент як Figma.

Figma - крос-платформний онлайн-сервіс для дизайнерів інтерфейсів і веб-розробників. Розробка інтерфейсів відбувається в онлайн-додатку.

У Figma дві ключові особливості: доступ до макету прямо з вікна браузера і можливість спільної роботи над документами.

3.1.1 Можливості та особливості Figma

Розрахований на багато користувачів режим редагування.

До появи Figma кільком дизайнерам складно було працювати над одним проектом і передавати макети розробникам. Photoshop відмовлявся коректно відкривати макет, поки ви не встановите потрібні шрифти. Або колега вносив зміни в свою копію проекту і забував сказати вам про це.

Команда Figma врахувала подібні проблеми і створила продукт, який дозволяє працювати над проектом одночасно кільком дизайнерам, підтримує версійність і дає багато інших можливостей. Вони спрощують життя як дизайнеру, так і розробнику.

Хмарний сервер зберігання файлів. Всі файли зберігаються в хмарі Figma. Не треба згадувати, де лежать ваші макети, завантажувати і після того, як внесли зміни, заливати їх назад. Зміни в Figma зберігаються автоматично.

Компоненти. Це елементи призначеного для користувача інтерфейсу, яким можна задавати загальні стилі - і потім швидко міняти їх у всьому макеті відразу.

Наприклад, ви закінчили розробку 50 макетів в Photoshop, замовник попросив вас поміняти пункт меню в шапці і надіслати все макети на

затвердження. Раніше доводилося проробляти це вручну. У Figma проблема вирішена: змінюєте щось одне - змінюються стилі всіх елементів з цим стилем.

Історія версій. У Figma легко переглядати ранні версії файлу і відновлювати або дублювати будь-яку з них. Версії автоматично зберігаються, якщо протягом 30 хвилин у файлі не було ніяких змін.

Ви бачите, коли файл був відредагований і хто зробив зміни. Можна і вручну додавати нові версії при необхідності. Щоб не заплутатися в подальшому, осмислено називайте версії і додавайте опис. Наприклад, деталі проекту: ітерацію, етап, які зміни внесли.

Фрейми. Дуже схожі на артборди в Photoshop, але мають велику гнучкість в роботі: вони добре масштабуються і їх з легкістю можна використовувати в інших фреймах. Для фреймів є встановлені розміри пристроїв: Phones, Tablet, Desktop, Watch, Paper, Social Media. Тому немає потреби тримати їх в голові і плутатися в розмірах.

Фрейм iPhone X і iPhone 8

Сітки. Панель Grid Layout дозволяє швидко створити будь-яку сітку.

- для одного макета необмежену кількість сіток;
- сітки швидко і гнучко налаштовуються, фіксуються і розтягуються;
- при необхідності колір сітки легко поміняти.

Десктопна версія. Figma - онлайн-сервіс, але є і десктопна версія. Для роботи в ній потрібно авторизуватися і мати постійне інтернет-підключення. Файли зберігаються локально і при необхідності імпортуються в браузерну версію.

Це далеко не всі можливості програми. Дизайнеру важливі швидкість роботи над проектом і комунікація в процесі. Ці та багато інших завдань дозволяє вирішити онлайн-сервіс Figma.

3.2 Розробка макета програми

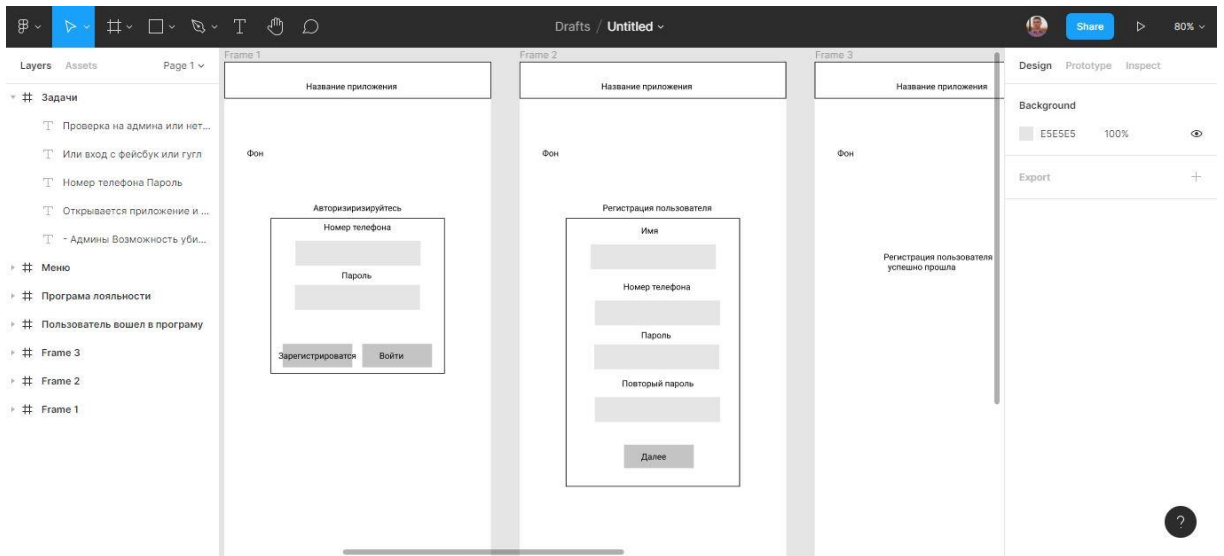


Рисунок 3.2.1 Розробка макета програми

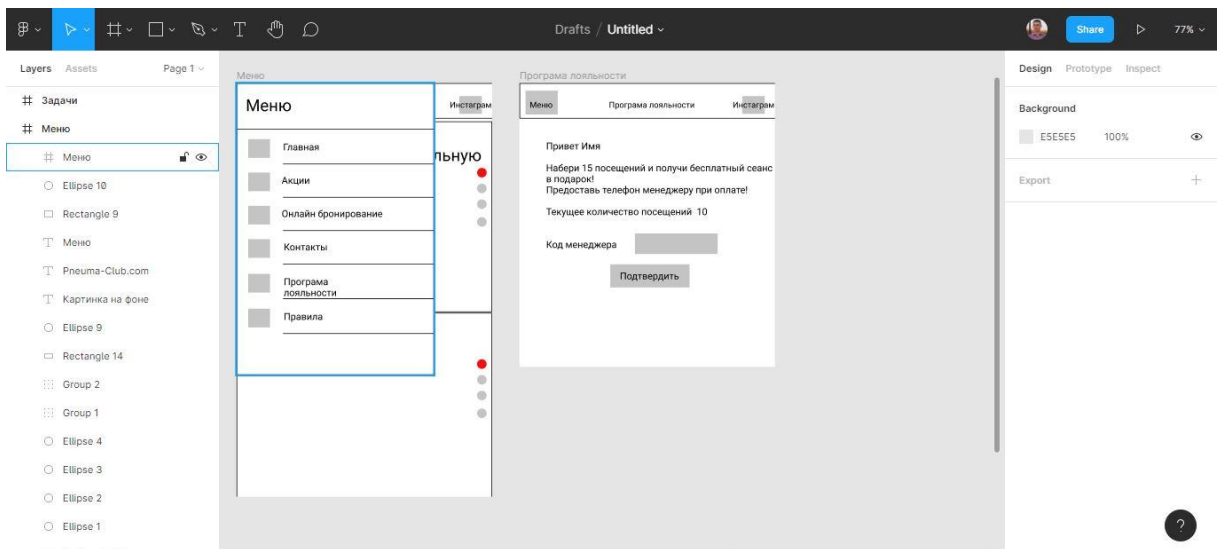


Рисунок 4.2.2 Розробка макета програми

4 СЕРЕДОВИЩЕ РОЗРОБКИ

За середовище розробки я обрав **Visual Studio Code** .

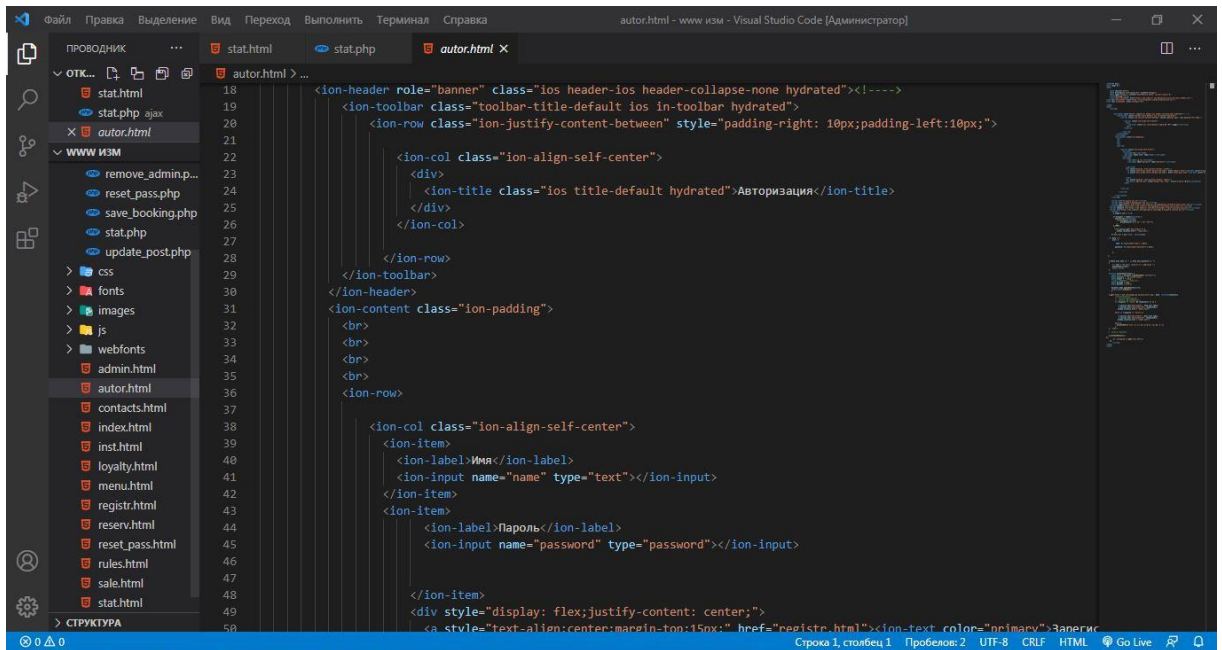


Рисунок 4.1 Середовище розробки Visual Studio Code

Visual Studio Code — засіб для створення, редагування та зневадження сучасних веб-застосунків і програм для хмарних систем. Visual Studio Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux і OS X.

Компанія Microsoft представила Visual Studio Code у квітні 2015 на конференції Build 2015. Це середовище розробки стало першим крос-платформовим продуктом у лінійці Visual Studio.

За основу для Visual Studio Code використовуються напрацювання вільного проекту Atom, що розвивається компанією GitHub. Зокрема, Visual Studio Code є надбудовою над Atom Shell, що використовують браузерний рушій Chromium і Node.js. Примітно, що про використання напрацювань вільного проекту Atom на сайті Visual Studio Code і в прес-релізі і в офіційному блозі не згадується.

4.1 Особливості Visual Studio Code

VS Code дозволяє розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ.

У редакторі присутні вбудований відладчик, інструменти для роботи з Git і засоби рефакторинга, навігації по коду, автодоповнення типових конструкцій і контекстної підказки.

Продукт підтримує розробку для платформ ASP.NET і Node.js, і вважається легковажним рішенням, яке дозволяє обійтися без повної інтегрованого середовища розробки.

Великим плюсом редактора є підтримка великої кількості мов, таких як JavaScript, C++, C#, TypeScript, jade, PHP, Python, XML, Batch, F#, DockerFile, Coffee Script, Java, HandleBars, R, Objective-C, PowerShell, Luna, Visual Basic, Markdown, JSON, HTML, CSS, LESS і SASS, Нахе.

4.2 Можливості Visual Studio Code

- вбудовані інструменти інтеграції з GitHub, GIT, а також Visual Studio Team Services для швидкого тестування, складання, упаковки та розгортання різних типів додатків;
- зручність роботи з Unity-проектами;
- робота з Mono і Node.js за допомогою вбудованого відладчика;
- підтримка TypeScript і JavaScript;
- публікація створених додатків в Microsoft Azure через сервіс Visual Studio Team Services;
- підтримка практично всіх мов програмування;
- написання коду для конкретного завдання з його подальшою інтеграцією в проект (з надбудовою або безпосередньо);

- велика бібліотека шаблонів, готових фрагментів коду і сніпетів з можливістю додавання своїх елементів;
- одночасна робота з декількома проектами (в декількох вікнах);
- інтерфейс можна розділити на дві панелі для порівняння коду;
- функція налагодження.

4.3 Переваги Visual Studio Code

- безліч налаштувань (як всієї програми, так і інтерфейсу);
- розширювана бібліотека доповнень і готових рішень;
- мультифункціональність (редактор підтримує майже всі мови, які використовуються для створення додатків);
- простота і гнучкість.

VS Code поширюється безкоштовно і розробляється, як програмне забезпечення з відкритим вихідним кодом.

4.4 Емулятор Genymotion

Всі, хто займається розробкою додатків під Android, я думаю, помічали, що емулятор віртуального андроїд-пристрої страждає серйозними проблемами з продуктивністю. Час завантаження і швидкість роботи віртуального пристрою навіть на потужних машинах просто виходять за всякі розумні рамки. Іноді здається, що емулятор не завантажили ніколи.

Що ж робити бідному андроїд розробнику? Вихід є! Це Genymotion - рішення, засноване на Virtual Box, яке виртуалізує, а не емулює Android, забезпечуючи відчутний приріст продуктивності. Genymotion має повний набір засобів для швидкої і якісної роботи: shell, більше 10 віртуальних пристроїв і інтеграцію з Eclipse і Android Studio. Причому все це працює на Linux, Mac і Windows і має зручний інтерфейс.

На сайті genymotion.com нам пропонується три варіанти установки - два з них платні (99 і 299 євро в рік відповідно), і один безкоштовний. Цей не включає преміум підтримку і не підтримує деякі функції, доступні в платних версіях, наприклад, захоплення відео з екрану пристрою. Але для тестування додатків його більш ніж достатньо.

Genymotion - це один з кращих емуляторів Андроїд на ПК. Він відрізняється швидкою роботою і дозволяє зручно перевіряти працездатність додатків. Емулятор Genymotion є вдосконаленим продовженням AndroidVM. Він адресований широкому колу користувачів і може бути використаний в «тандемі» з операційними системами Windows, MacOS і Linux.

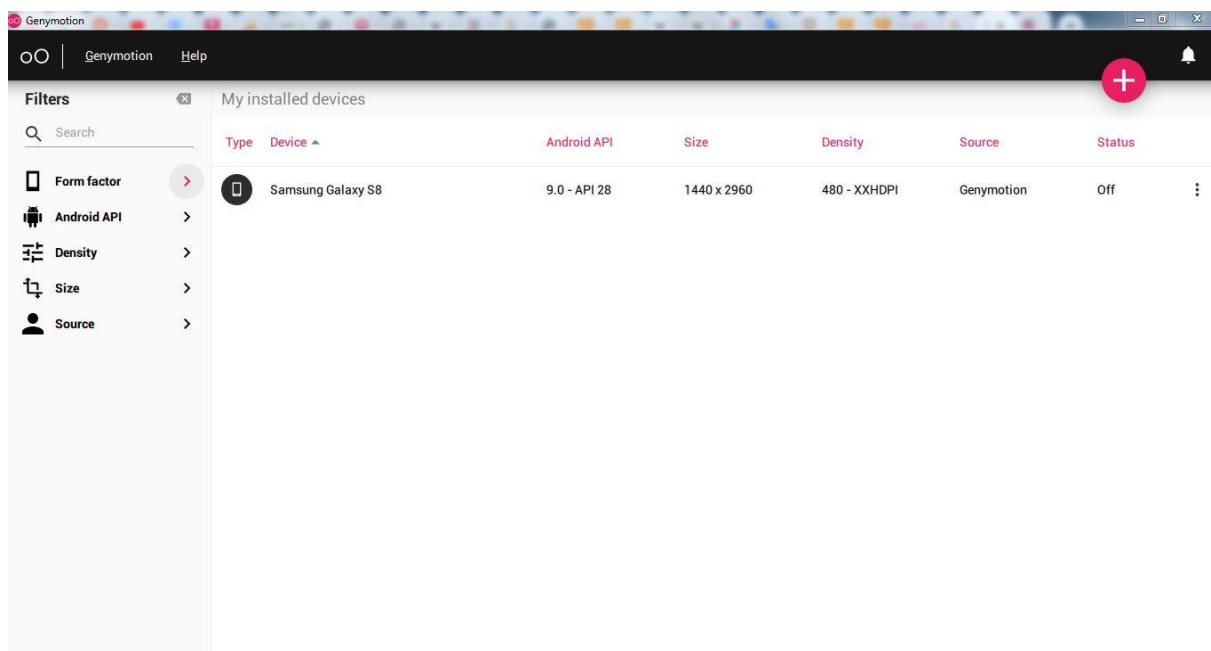


Рисунок 4.4.1 Знімок мого емулятора



Рисунок 4.4.2 Знімок робочого столу мого емулятора

4.5 Особливості Genymotion

Особливості Genymotion:

- Швидке завантаження емулятора.

- Миттєвий запуск образів Android, які вже були налаштовані.
- Можливість емуляції бездротового Wi-Fi з'єднання.
- Емуляція інтернету за допомогою різних технологій.
- Наявність режиму повноекранного перегляду.
- Підтримка ADB.
- Підтримка трьох ОС: Windows, Mac і Linux.
- Можна завантажувати додатки з Google Play.
- Спалах і камери можуть емулюватися.

І це ще не все особливості такого емулятора. Також варто відзначити наявність плагінів до Eclipse. Цей емулятор відрізняється найшвидшою роботою серед всіх подібних програм.

У платній версії емулятора доступна можливість заміни IMEI віртуального пристрою. Також її користувачі можуть при необхідності повністю скинути налаштування віртуального пристрою на операційній системі Android. Для цього передбачена настройка Factory Reset.

4.6 Недоліки емулятора Genymotion

Недоліки емулятора Genymotion:

- Відсутня інтерфейс українською та російською мовою.
- Голосові дзвінки, не емулюються.

Для початку роботи необхідно пройти не складну реєстрацію, після чого завантажити і встановити сам Андроїд емулятор Genymotion і можна приступати до емуляції додатків Android на комп'ютері.

5 ТЕХНОЛОГІЇ ЯКІ Я ВИКОРИСТОВУВАВ

5.1 Фреймворк інтерфейсу Ionic Framework

Якщо інтерфейс – це взаємодія людини з неживим предметом, то користувацький інтерфейс – це взаємодія людини і комп'ютера: сайти, мобільні додатки, програми. І цю взаємодію має хтось проектувати. Цим займаються дизайнери інтерфейсів, яких ще називають UI/UX-дизайнерами. Вони працюють над принципами роботи системи, послідовністю дій, які може здійснити користувач, результатом, який він отримає на виході, зрозумілістю, красою і зручністю об'єкта в користуванні. Мета роботи дизайнера інтерфейсів – зробити взаємодію людини та програми приємною, логічною і дружньою. Це робота на перетині дизайну, інженерії, маркетингу і психології.

Зручність та красу інтерфейсів часто проектує одна й та сама людина, але інтерфейси стають все складнішими, тому професію ділять на дві. User Interface Designer (UI) – той, хто відповідає за красу та задоволення. User Experience Designer (UX) – той, хто відповідає за зручність та відповідність бізнес-задачам.

UI-дизайнер займається усім, що стосується оформлення інтерфейсу і створює зрозумілі, цілісні та гарні інтерфейси для користувача. Його ключові обов'язки: розробка стилю, створення макетів, безпосередній дизайн сторінок. Він працює з кольорами, іконками, типографією, навігацією, меню, кнопками, вікнами, анімацією, сповіщеннями. UI-фахівець створює дизайн, базуючись на даних, отриманих від UX-спеціаліста.

UX-дизайнер вивчає проблеми користувача, розбирається у його поведінці, досліджує досвід. UX-дизайнер повинен впевнитися, що продукт працює логічно та вирішує конкретні проблеми. Ключові обов'язки UX-профі: дослідження аудиторії та продукту, проектування користувацьких сценаріїв. UX-дизайнер займається «щастям» користувача: задоволенням і продуктивністю від роботи з інтерфейсом, загальним розумінням і легкістю вирішення проблем.

Ролі обох дизайнерів перетинаються, тому займатися лише UI неможливо без знань UX – і навпаки.

Щоб полегшити роботу при створенні додатків вже є готові фреймворки інтерфейсів. Один з таких Ionic Framework.

Ionic - це повний SDK з відкритим кодом для розробки гібридних мобільних додатків, створений Максом Лінчем, Беном Сперрі та Адамом Бредлі з Drifty Co. у 2013 році. Оригінальна версія випущена в 2013 році і побудована на базі AngularJS та Apache Cordova. Однак останній випуск був перероблений як набір веб-компонентів, що дозволило користувачеві вибрати будь-яку структуру інтерфейсу користувача, наприклад Angular, React або Vue.js. Це також дозволяє використовувати іонічні компоненти, що взагалі не мають інтерфейсу користувача. Ionic надає інструменти та послуги для розробки гібридних мобільних, настільних та прогресивних веб-програм, заснованих на сучасних технологіях та практиці веб-розробки, використовуючи веб-технології, такі як CSS, HTML5 та Sass. Зокрема, мобільні програми можна створювати за допомогою цих веб-технологій, а потім розповсюджувати їх у власних магазинах додатків, щоб встановлювати на пристрої за допомогою Cordova або Capacitor.

Послуги та функції. Ionic використовує Cordova та нещодавно плагіни Capacitor, щоб отримати доступ до таких функцій операційної системи, як камера, GPS, ліхтарик тощо. Користувачі можуть створювати свої програми, а потім їх можна налаштувати для Android , iOS , Windows , робочий стіл (з Electron) або сучасні браузері. Ionic дозволяє створювати та розгортати програми, обгортаючи інструмент побудови Cordova або Capacitor за допомогою спрощеного інструмента командного рядка.

Ionic включає мобільні компоненти, типографіку, інтерактивні парадигми та розширювану базову тему.

Використовуючи веб-компоненти, Ionic надає власні компоненти та методи взаємодії з ними. Один з таких компонентів, віртуальна прокрутка, дозволяє користувачам прокручувати список тисяч елементів без будь-яких звернень

продуктивності. Інший компонент, вкладки, створює інтерфейс із вкладками з підтримкою навігації та управління станом історії.

Окрім SDK, Ionic також надає послуги, які розробники можуть використовувати для включення таких функцій, як розгортання коду , автоматизовані збірки . Ionic також пропонує власну IDE, відому як Ionic Studio.

Ionic також надає інтерфейс командного рядка (CLI) для створення проєктів. CLI також дозволяє розробникам додавати плагіни Cordova та додаткові інтерфейсні пакети, увімкнути push-сповіщення, генерувати піктограми програм та заставки та створювати власні двійкові файли.

Підтримувані платформи.Для Android Ionic підтримує Android 4.4 та новіших версій. Для iOS Ionic підтримує iOS 10 і новіших версій. Ionic 2 підтримує універсальну платформу Windows для створення програм для Windows 10 . Ionic Framework, заснований на Angular.js , підтримує програми BlackBerry 10

Продуктивність.Додатки Ionic працюють із сумішню власного коду та веб-коду, забезпечуючи повний доступ до власних функціональних можливостей, якщо це необхідно, з основним інтерфейсом програми, побудованим за стандартною веб-технологією. Ionic використовує власні функції апаратного прискорення, доступні в браузері (наприклад, анімація CSS), і оптимізує візуалізацію (уникаючи дорогих маніпуляцій з DOM). Ionic використовує переходи та перетворення CSS для анімації як спосіб використовувати графічний процесор і максимізувати доступний час процесора.

Ionic - це технологія, що дозволяє розробляти повноцінні програми для iOS і Android. Для цього не потрібно мати глибокі знання в кожній з платформ. Звичайно ж, є деякі обмеження, але в цілому необхідно бути знайомим з Angular (популярний веб-фреймворк), щоб почати розробку програми. Для застосування стилів можна використовувати SCSS - це додасть додатком потрібний вид. Розглянемо основні переваги та недоліки Ionic.

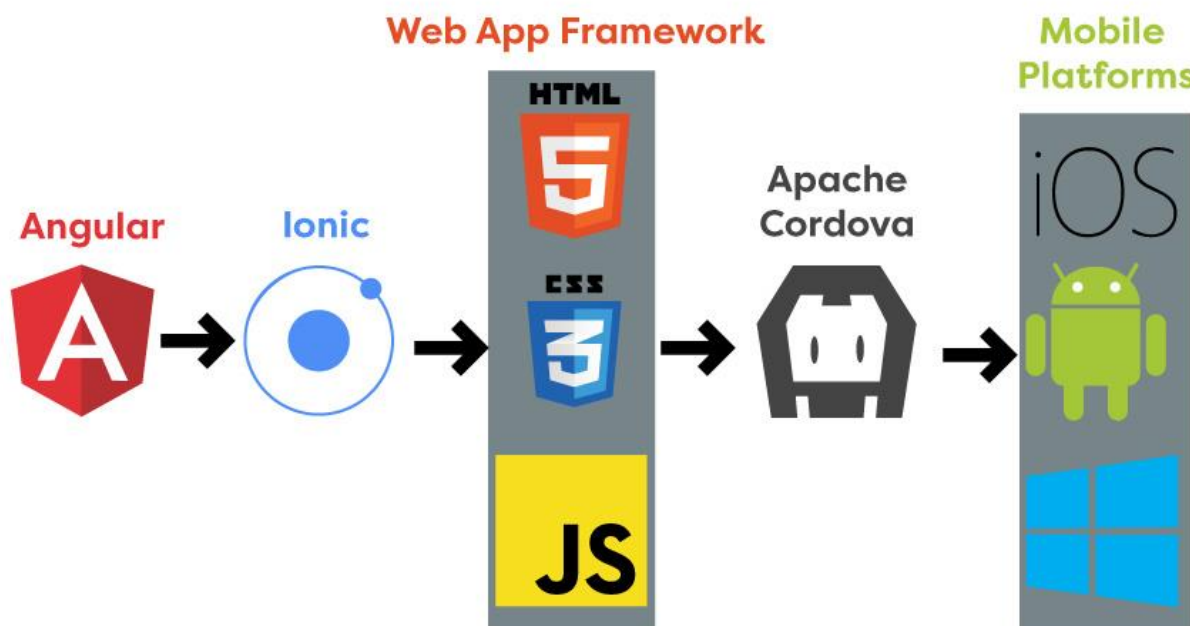


Рисунок 5.1.1 Схема створення мобільного додатку з Ionic

У Ionic є вбудована бібліотека стандартних елементів, які можна використовувати аналогічно елементам Bootstrap: картки, кнопки, перемикачі, сегменти, попап, поля введення, списки, сітка з рядків і колонок і т. д. За замовчуванням ці елементи змінюються так, щоб виглядати як нативні на iOS і Android, але їх вид можна змінити і при необхідності.

Також з Ionic вам доступна безліч плагінів, які дозволяють використовувати залізо смартфона (Ionic Native / Cordova). Але не забудьте простежити, щоб ваші платформи активно підтримували вибрані плагіни. У більшості випадків такі плагіни працюють добре, але іноді може виникнути помилка билда або конфлікт навіть з широко використовуваними плагінами начебто входу через Facebook або Firebase Analytics. Такі проблеми зазвичай вирішуються або чищенням і ребілд проекту, або оновленням плагіна (за умови, що вже є його нова версія, яка вирішує проблему). Також можна замінити плагін на альтернативний або в деяких випадках - додати параметри, що дозволяють перезапис в `config / xml /`.

У Ionic є ще один великий плюс - швидкість розробки. Оскільки він заснований на Angular, проект на Ionic можна запускати в браузері і бачити, як буде виглядати додаток під час розробки. Для того щоб побачити таке превью, не

обов'язково встановлювати додаток на смартфон або емулятор. Це істотно допомагає економити час при зміні UI. А коли ви працюєте над функціями, які вимагають перевірки на смартфоні (наприклад, зробити фото), збірка билда і установка його на смартфон займе всього кілька хвилин. На Android встановлювати додатки можна прямо з командного рядка, а на iOS білд потрібно відкрити в Xcode.

Гібридні vs нативні додатки.Нещодавно з'явилося кілька фреймворків, що дозволяють розробляти справжні нативні додатки з використанням Angular (NativeScript) або React (React Native). Вони мають важливу перевагу перед гібридним підходом Ionic - ніщо не може зрівнятися з продуктивністю нативних додатків. Однак нативні додатки мають не менш важливий мінус. Використовуючи Ionic, ви працюєте з HTML- і SCSS-файлами, а для нативної розробки вам знадобляться інші навички для роботи з розміткою і стилями. Цих навичок немає у більшості веб-розробників.

Деплой в Google Play Store.Щоб випустити завершене додаток, вам знадобиться його спочатку зібрати:

`ionic cordova build android --prod --release`. Потім його потрібно підписати за допомогою `jarsigner` (інструмент для підпису APK), використовуючи свій JKS-ключ і `zipalign` (інструмент для перетворення підписаного додатки в APK в готове для завантаження на Google Play Store). При цьому важливо використовувати той же JKS-ключ, який завантажений в вашу консоль розробника Play Store.

iOS.Ionic для iOS не так налагоджений, як для Android.Сегмент - іноді він може конфліктувати зі скролінгом. Також багато елементів вимагають окремих SCSS- стилів для того, щоб вони виглядали як вам потрібно на обох платформах. Те ж відноситься до плагіну Flurry Analytics. Його легко встановити і використовувати в Android, а на iOS цей плагін не встановлювався простими способами і зажадав використовувати CocoaPods - пакетний менеджер для установки плагінів на iOS. І навіть після правильної установки цей доданок не ініціалізується так само, як на Android.

Інший плагін, який перетворює зображення в base64, видає злегка відрізняється формат base64 на Android і iOS. Але в цілому більшість плагінів повинні працювати, хоча деякі вимагають тонких налаштувань.

Плюси:

- Швидка розробка і мінімальний час виходу на ринок.
- Можна вести основну частину розробки в браузері (крім нативної функціональності смартфона - тут знадобиться використовувати смартфон для дебага).
- Можна розробляти додаток для iOS і Android одночасно (з деякими обмеженнями - такими як особливості платформ, що стосуються стилів і плагінів).
- Навички в Angular, HTML, CSS і JavaScript - це практично все, що знадобиться для початку розробки. Немає необхідності знати Java і Swift або Objective-C.
- Безліч UI-компонентів доступні і прості у використанні - картки, кнопки, перемикачі, сегменти, попап, поля введення, списки, сітка з рядків і колонок і т. д.
- Безліч плагінів, що дозволяють використовувати функції смартфонів, такі як: камера, сканер відбитків пальців, NFC, геолокація, відправка аналітики на Firebase, оповіщення та глибинні посилання.

Мінуси:

- Нативні плагіни можуть стати проблемою, якщо якісь із використовуваних плагінів конфліктують або якщо в одному з них баг. Наприклад, недавно в плагіні для логіна за допомогою Facebook був баг: якщо юзер хоч раз зробив log out, то більше логін вже не працював. А плагін FCM (Firebase Cloud Messaging) не працює з firebase-analytics. Але є плагін під назвою Firebase, яким можна замінити їх обидва. Дебаг може бути досить складним: іноді важко зрозуміти, звідки приходить помилка, оскільки повідомлення про помилки можуть бути неінформативними.

- Білд може ламатися без причин, щось виявляється пошкодженим в оригінальній папці. Так що робіть коммітов часто і використовуйте гілки для кожної нової функції або сторінки. Якщо щось зламається, просто зробіть клон сховища в новій папці, запустіть `npm install` і спробуйте зібрати проект заново.

Приклад використання Фреймворку в меню програми

```
<ion-menu side="start" content-id="main-
content" role="navigation" class="ios menu-type-overlay menu-enabled menu-side-
start hydrated">
  <ion-header role="banner" class="ios header-ios header-collapse-
none hydrated">
    <ion-toolbar translucent="" class="toolbar-title-default ios in-
toolbar hydrated">
      <ion-title class="ios title-default hydrated">Menu</ion-title>
    </ion-toolbar>
  </ion-header>
  <ion-content class="ios hydrated" style="--offset-top:0px; --offset-
bottom:0px;">
    <ion-list class="ios list-ios hydrated">
      <ion-item class="item ios in-list ion-focusable hydrated item-label">
        <a class="none_decor" href="index.html"><ion-label class="sc-ion-label-
ios-h sc-ion-label-ios-s ios hydrated">Главная</ion-label></a>
      </ion-item>
      <ion-item class="item ios in-list ion-focusable hydrated item-label">
        <a class="none_decor" href="sale.html"><ion-label class="sc-ion-label-
ios-h sc-ion-label-ios-s ios hydrated">Акции</ion-label></a>
      </ion-item>
```

```
<ion-item class="item ios in-list ion-focusable hydrated item-label">
```

```
  <a class="none_decor" href="reserv.html"><ion-label class="sc-ion-label-  
ios-h sc-ion-label-ios-s ios hydrated">Онлайн бронирование</ion-label></a>
```

```
</ion-item>
```

```
<ion-item class="item ios in-list ion-focusable hydrated item-label">
```

```
  <a class="none_decor" href="loyalty.html"><ion-label class="sc-ion-label-  
ios-h sc-ion-label-ios-s ios hydrated">Програма лояльности</ion-label></a>
```

```
</ion-item>
```

```
<ion-item class="item ios in-list ion-focusable hydrated item-label">
```

```
  <a class="none_decor" href="contacts.html"><ion-label class="sc-ion-  
label-ios-h sc-ion-label-ios-s ios hydrated">Контакты</ion-label></a>
```

```
</ion-item>
```

```
<ion-item class="item ios in-list ion-focusable hydrated item-label">
```

```
  <a class="none_decor" href="rules.html"><ion-label class="sc-ion-label-  
ios-h sc-ion-label-ios-s ios hydrated">Правила</ion-label></a>
```

```
</ion-item>
```

```
<ion-item class="item ios in-list ion-focusable hydrated item-label">
```

```
  <a class="none_decor" id="exit" href="#"><ion-label class="sc-ion-label-  
ios-h sc-ion-label-ios-s ios hydrated">Выйти</ion-label></a>
```

```
</ion-item>
```

```
</ion-list>
```

```
</ion-content>
```

```
</ion-menu>
```

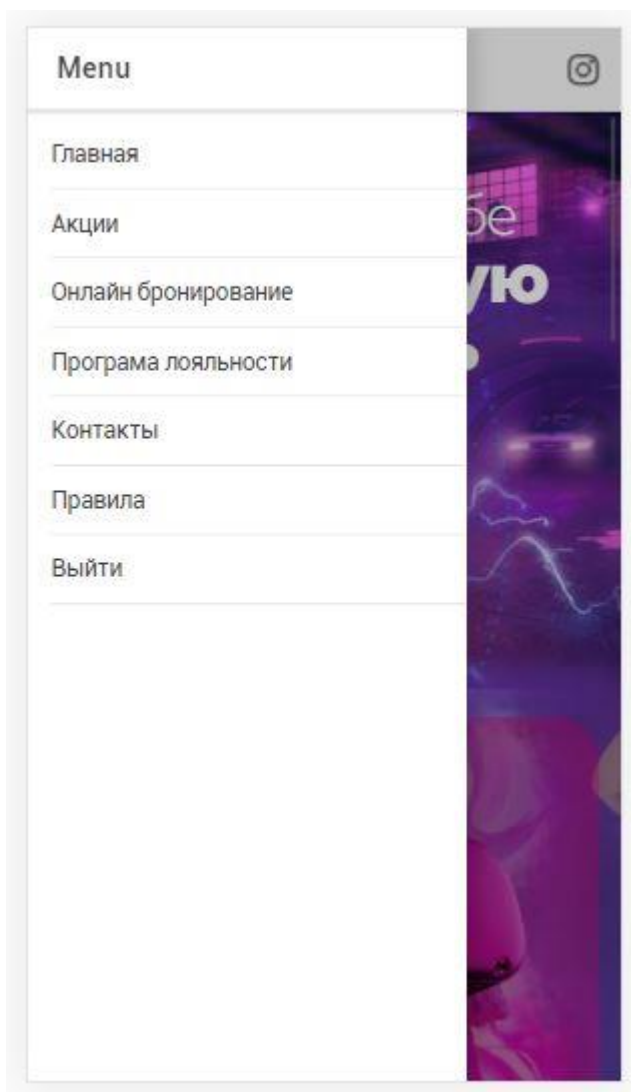


Рисунок 5.1.2 Результат виконання коду

5.2 Мова програмування Javascript

JavaScript (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну

та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить у браузерях як мова сценаріїв для надання інтерактивності веб-сторінкам.

Основні архітектурні риси:

- динамічна типізація,
- автоматичне керування пам'яттю,
- прототипне програмування,
- функції як об'єкти першого класу.

На JavaScript вплинули багато мов, при розробці була мета зробити мову схожою на Java, але при цьому легкою для використання непрограмістами. JavaScript має низку властивостей об'єктно-орієнтованої мови, але реалізоване в мові прототипування обумовлює відмінності в роботі з об'єктами в порівнянні з традиційними об'єктно-орієнтованими мовами. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам, - функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання - що додає мові додаткову гнучкість.

Незважаючи на схожий з Сі синтаксис, JavaScript у порівнянні з мовою Сі має корінні відмінності:

- об'єкти, з можливістю інтроспекції;
- функції як об'єкти першого класу;
- автоматичне приведення типів;
- автоматичне прибирання сміття;
- анонімні функції.
- У мові відсутні такі корисні речі, як:
 - модульна система - JavaScript не надає можливості управляти залежностями та ізоляцією областей видимості;

- стандартна бібліотека - зокрема, відсутній інтерфейс програмування додатків по роботі з файловою системою, управління потоками вводу/виводу, базових типів для бінарних даних;
- стандартні інтерфейси до веб-серверів та баз даних;
- система управління пакетами, яка б відстежувала залежності і автоматично встановлювала їх.

5.2.1 Актуальність

JavaScript, наразі, є однією з найпопулярніших мов програмування в інтернеті. В перші роки існування, більшість професійних програмістів скептично ставилися до мови, цільова аудиторія якої складалася з програмістів-аматорів. Поява AJAX змінила ситуацію та звернула увагу професійної спільноти до мови, а її подальші модифікації за стандартами ES6+ внесли багато корисних можливостей, яких не вистачало для ефективного програмування. В результаті, були розроблені та покращені багато практик використання JavaScript (зокрема, тестування та налагодження), створені бібліотеки та фреймворки, поширилося використання JavaScript поза браузером.

5.2.2 Опис

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 кілька 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 збирання сміття
- анонімні та стрілочні функції

JavaScript містить декілька десятків вбудованих об'єктів[17], які поділяються на групи: фундаментальні (Object, Function, Boolean, Symbol), помилки (група об'єктів Error), числа та дати (Number, BigInt, Math Date), текстові (String, RegExp), індексовані (група об'єктів Array), ключові (Map, Set, WeakMap, WeakSet), для роботи з структурованими даними (ArrayBuffer, Atomics, DataView, JSON), абстрактні (Promise, Generator), рефлексійні (Reflect, Proxy), групи Intl та WebAssembly. Крім того, JavaScript містить набір вбудованих операцій, що керують логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови Java (тобто, зрештою, успадкований від C), але спрощений у порівнянні з ним, щоб зробити мову сценаріїв легкою для вивчення. Так, наприклад, декларація змінної не містить її типу, властивості також не мають типів, а декларація функції може знаходитися в тексті програми після неї.

Одна з популярних технологій, що дозволила зробити сторінки динамічнішими і забезпечити нові можливості — це динамічне завантаження і вставка даних в документ, що отримала назву AJAX.

5.2.3 Використання в HTML

При використанні в рамках технології DHTML JavaScript код включається в HTML-код сторінки і виконується інтерпретатором, вбудованим в браузер. Код JavaScript вставляється в теги `<script></script>`, хоча в більшості браузерів мова сценаріїв за умовчанням саме JavaScript.

5.2.4 Типізація

Оскільки JavaScript є інтерпретованою мовою програмування, без строгої типізації, і може виконуватися в різних середовищах, кожне зі своїми власними особливостями сумісності, програміст має бути уважним, і повинен перевіряти, що його код виконується як очікується в широкому переліку можливих конфігурацій. Типізація вважається одною з ключових проблем JavaScript, тому восени 2012 року, компанія Microsoft презентувала мову програмування TypeScript, що компілюється в JavaScript та містить декілька важливих для програмістів доповнень, що полегшують розробку.

5.2.5 Відлагодження

При розробці великих і нетривіальних вебзастосунків з використанням JavaScript, критично важливим є доступ до інструментів відлагодження. Оскільки браузері, від різних виробників, дещо відрізняються у поведінці JavaScript і реалізації Об'єктної моделі документа, необхідно мати відлагоджувач для кожного браузера, якщо вебзастосунок орієнтовано на нього.

На даний час Firefox, Opera, Google Chrome, Edge та Safari мають зневаджувачі для себе.

Також існують такі корисні інструменти, як:

- ESLint — перевірка якості коду, що сканує JavaScript-програму, шукаючи вади у коді;
- Prettier — автоматичне форматування коду у коректний вигляд;
- Babel — компілятор JavaScript-коду до старіших версій стандарту ECMAScript, який допомагає розробникам використовувати найновіші можливості мови для оточення, що не встигло реалізувати останній стандарт.

Кожен блок сценарію інтерпретатор розбирає окремо. На вебсторінках, коли треба комбінувати блоки JavaScript та HTML, синтаксичні помилки знайти легше, якщо зберігати функції сценарію в окремому блоці коду, або (ще краще) використовувати багато малих, пов'язаних між собою .js файлів. Таким чином,

синтаксична помилка не спричинятиме «падіння» всієї вебсторінки та дозволить сповістити користувача про проблему.

Приклад коду JavaScript

Наведений нижче код показує дату та час при натисканні на кнопку.

```
<!DOCTYPE html>
<html>
<body>

<h2>My First JavaScript</h2>

<button type="button"
onclick="document.getElementById('demo').innerHTML = Date()">
Click me to display Date and Time.</button>

<p id="demo"></p>

</body>
</html>
```

5.3 Бібліотека jQuery

jQuery — популярна JavaScript-бібліотека з відкритим кодом. Вона була представлена у січні 2006 року у BarCamp NYC Джоном Ресігом (John Resig). Згідно з дослідженнями організації W3Techs, JQuery використовується понад половиною від мільйона найвідвідуваніших сайтів. jQuery є найпопулярнішою бібліотекою JavaScript, яка посилено використовується на сьогоднішній день.

jQuery є вільним програмним забезпеченням під ліцензією MIT (до вересня 2012 було подвійне ліцензування під MIT та GNU General Public License другої версії).

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 дана бібліотека 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.

```

$( '#test' ) //знаходимо елемент з id="test"
    .text('Клікни по мені') //встановлюємо текст елемента рівним "Клікни по мені"
    .addClass('myAlert') //додаємо клас "myAlert"
    .css('color','red') //встановлюємо колір тексту червоним
    .attr('alert','Привіт, світе!') // додаємо атрибут "alert" із значенням "Привіт, світе!"
    .bind( // додаємо в обробник події click функцію, яка відкриє модальне
        'click', // вікно із текстом, що вказаний в атрибуті "alert"
        ("Привіт, світе!")
        function(){ alert($(this).attr('alert'))}
    );

```

Приклад файла registration.js з програми

```

$( document ).ready(function() {

```

```

$( "#num" ).focus(function() {
    $("#item").addClass("item-interactive item-input item md ion-
focusable hydrated item-label item-has-focus item-has-value")
});
$('#num').focusout(function(){
    $("#item").removeClass("item-has-focus")
});
$('#num').mask('+38 (099) 999-99-99');
$('#reg_but').on('click', function(e){

var data = {
    user: {

        name: $('input[name="name"]').val(),
        email: $('input[name="email"]').val(),
        phone: $('input[name="number"]').val(),
        password: $('input[name="password"]').val(),
        password2: $('input[name="password2"]').val()

    },

};

    if(data.user.name == "" || data.user.phone == "" || data.user.password == "" || data.
user.password2 == "" || data.user.email == "")
    {
        var text = 'Все поля обязательны к заполнению!';
        presentAlert(text);
        return false;
    }
}

```

```
}  
if(data.user.password !== data.user.password2)  
{  
  
    var text = 'Пароли не совпадают!';  
    presentAlert(text);  
    return false;  
}  
if(data.user.password.length < 8)  
{  
  
    var text = 'Пароль слишком короткий(минимум 8 символов)!';  
    presentAlert(text);  
    return false;  
}  
var pattern = /^[a-z0-9_-]+@[a-z0-9-]+\.[a-z]{2,6}$/i;  
if(data.user.email.search(pattern) !== 0)  
{  
    var text = 'Адрес почты введен неверно!';  
    presentAlert(text);  
    return false;  
}  
  
function presentAlert(text) {  
    const alert = document.createElement('ion-alert');  
    alert.cssClass = 'my-custom-class';  
    alert.header = 'Ошибка';  
    alert.message = text;  
    alert.buttons = ['OK'];
```

```
document.body.appendChild(alert);
return alert.present();
}

$.post('https://www.testingapp.pp.ua/ajax/add_user.php', data, function(response)
{
    alert(response);

    setTimeout(function(){
        window.location.href = "autor.html";
    }, 3000);

}, 'json')

console.log(data);

e.preventDefault();
});
$('.instagram').load("inst.html");
});
```

5.4 Мова програмування PHP

PHP (англ. PHP: Hypertext Preprocessor — PHP: гіпертекстовий препроцесор), попередня назва: Personal Home Page Tools — 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 — проєкт 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, тому що браузер отримує готовий html-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніхто не забороняє використовувати PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта.

PHP — мова, у код якої можна вбудовувати безпосередньо html-код сторінок, які, у свою чергу, коректно оброблюватимуться PHP-інтерпретатором. Обробник PHP просто починає виконувати код після відкриваючого тегу (`<?php`) 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.

5.4.1 Наявність інтерфейсів до багатьох баз даних

- у PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, SQLite, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase.
- завдяки стандарту відкритого інтерфейсу зв'язку з базами даних (англ. Open Database Connectivity Standard, ODBC) можна підключатися до всіх баз даних, до яких існує драйвер.

5.4.2 Нетрадиційність

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 мовами C або Pascal. Це помітно знижує початкові зусилля при вивченні PHP. PHP — мова, що поєднує переваги Perl та 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 найпопулярнішою мовою для створення веб-застосунків (скриптів).

5.4.3 Наявність сирцевого коду та безкоштовність

Стратегія Open Source, і розповсюдження початкових текстів програм в масах, безсумнівно справили сприятливий вплив на багато проєктів, в першу чергу — Linux хоч і успіх проєкту 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 проєкту PHP. Ухвалення стратегії Open Source і безкоштовне 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.

5.4.4 Ефективність

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.

5.4.5 Додаткові можливості

Мова явно підтримує HTTP cookies відповідно до специфікацій Netscape. Це дозволяє проводити встановлення та читання невеликих сегментів даних на

стороні клієнта. PHP надає можливість організації роботи з користувачем протягом сеансів (сесій). В сесії можна зберігати різні дані, включаючи об'єкти.

Безпека:

- Засоби безпеки системного рівня. PHP можна налаштувати так, щоб вона забезпечувала максимальну свободу дій і безпеку. PHP може працювати в безпечному режимі (safe mode), який обмежує можливості застосування PHP користувачами. Наприклад: максимальний час виконання та використання пам'яті.

- Засоби безпеки рівня програми. PHP включає надійні механізми шифрування. PHP також сумісний з багатьма додатками інших розробників, що дозволяє легко інтегрувати його з захищеними технологіями електронної комерції. Вихідний код PHP не можна переглянути у браузері, оскільки він виконується на сервері.

Гнучкість. PHP використовується не лише у поєднанні з HTML, але й із JavaScript, WML, XML та іншими мовами програмування. PHP-код може передаватися будь-яким браузерам і пристроям, в тому числі стільниковим телефонам, портативним комп'ютерам. PHP-код можна виконувати в режимі командного рядка.

PHP працює на різних web-серверах (Apache, Netscape Enterprise Server, Microsoft IIS, Stronghold, Zeus) і платформах (UNIX, Solaris, FreeBSD, Windows 95/98/NT/2000/XP/2003).

Приклад файла autor.php з програми

Наведений нижче код реалізує авторизацію користувача в мобільний додаток. Він перевіряє чи є користувач з таким іменем та паролем в базі даних, якщо є то код передає запит на бік клієнта з id користувача, якщо такого користувача не існує то повертає false.

```
<?php
```

```
header("Access-Control-Allow-Origin: *");
```

```

if (isset($_GET['user'])) {
    $response = array();
    $name=htmlspecialchars($_GET['user']['name']);
    $password=htmlspecialchars($_GET['user']['password']);

    $p = sha1($password);
    $dbc = mysqli_connect("ue418458.mysql.tools", "ue418458_db", "23BqzvC
z", "ue418458_db");
    mysqli_set_charset($dbc, 'utf8');
    $query = "SELECT * FROM `users_aut` WHERE name = ".$name." AND
password = ".$p." ";
    $result = mysqli_query($dbc, $query);
    $row_cnt = mysqli_num_rows($result);
    if($row_cnt!=0)
    {

    $row=mysqli_fetch_assoc($result);
    $dbusername=$row['name'];
    $dbpassword=$row['password'];
    $userid=$row['id'];
    $user_admin=$row['admin'];
    if($name == $dbusername && $p == $dbpassword)
    { array_push($response, $userid);
      array_push($response, $user_admin);
      echo json_encode($response);
    }
    } else {
        echo json_encode("false");
    }
}

```

```

mysqli_close($dbc);
}

?>

```

5.5 Використання JSON серіалізації

Приклад використання JSON формату передачі даних з серверу на мові PHP для обробки мобільним додатком мовою Javascript з використанням бібліотеки JQuery.

```

<?php
    header("Access-Control-Allow-Origin: *");

    $response = array();

    $dbc = mysqli_connect("ue418458.mysql.tools", "ue418458_db", "23BqzvC
z", "ue418458_db");

    mysqli_set_charset($dbc, 'utf8');
    $query = "SELECT * FROM `users` WHERE done = '0' ORDER BY date A
SC";

    $result = mysqli_query($dbc, $query);
    $row_cnt = mysqli_num_rows($result);
    if($row_cnt!=0)
    {

    while($row=mysqli_fetch_assoc($result))
    {

        $name=$row['name'];
        $count_people=$row['count_people'];

```

```

    $phone=$row['phone'];
    $cerf=$row['cerf'];
    $id_user=$row['id_user'];
    $date=$row['date'];
    $time=$row['time'];
    $done=$row['done'];
    $dct = array("name" => $name ,"count_people" => $count_people ,"phone
" => $phone ,"cerf" => $cerf ,
    "id_user" => $id_user ,"date" => $date,"time" => $time,"done" => $done);
    array_push($response,$dct);
}

```

```

    echo json_encode($response);

```

```

} else {
    echo json_encode("false");
}

```

```

mysqli_close($dbc);

```

```

?>

```

```

function get_posts() {
    $.ajax({
        type: "POST",
        url: "https://www.testingapp.pp.ua/ajax/admin.php",
        cache: false,
        async:true,

```

```

dataType: 'json',
success: function(response){
  console.log(response);
  if (response != "false") {
    $('ion-content').empty();
    for (var i=0 ; i < response.length ; i++){
      $('ion-content').append(

        '<div class="wrap_card">'+
        '<ion-card class="main_card">'+
        '<ion-card-header>'+
        '<ion-card-title>Дата: '+ response[i].date +'</ion-card-title>'+
        '<ion-card-subtitle>Время: '+ response[i].time +'</ion-card-
subtitle>'+

        '</ion-card-header>'+

        '<ion-card-content>'+
        '<p id="id" style="visibility:hidden">'+ response[i].id_user +'<
/p>'+

        '<p>Имя: '+ response[i].name +'</p>'+
        '<p>Количество человек: '+ response[i].count_people +'</p>'

        +

        '<p>Мобильный номер: '+ response[i].phone +'</p>'+
        '<p>Номер сертификата: '+ response[i].cerf +'</p>'+
        '</ion-card-content>'+
        '<ion-button id="remove" color="danger">Удалить<ion-
icon style="padding-left: 10px;" name="trash-outline"></ion-icon>'+
        '</ion-button>'+
        '<ion-button id="change">Изменить<ion-icon style="padding-
left: 10px;" name="pencil-outline"></ion-icon></ion-button>'+

```

```

'</ion-card>'+

'<ion-card class="hidden_inner_card" style="display:none">'+
'<ion-card-header>'+
'<ion-card-
title>Дата: <input type="text" name="date" value="'+response[i].date+'"></ion-card-
title>'+

'<ion-card-
subtitle>Время: <input type="text" name="time" value="'+response[i].time+'"></ion-
card-subtitle>'+

'</ion-card-header>'+

'<ion-card-content>'+

'<p>Имя: <input type="text" name="name" value="'+response
[i].name+'"></p>'+

'<p>Количество человек: <input type="number" name="count_
people" value="'+response[i].count_people+'"></p>'+

'<p>Мобильный номер: <input type="tel" id="phone" name=
"phone" value="'+response[i].phone+'"></p>'+

'<p>Номер сертификата: <input type="text" max="5" name=
"cerf" value="'+response[i].cerf+'"></p>'+

'</ion-card-content>'+

'<ion-button id="back">Назад<ion-icon style="padding-
left: 10px;" name="arrow-undo-outline"></ion-icon>'+

'</ion-button>'+

'<ion-button id="send">Подтвердить<ion-icon style="padding-
left: 10px;" name="checkmark-outline"></ion-icon></ion-button>'+

'</ion-card></div>');

```



```

    }
    }else presentAlert("Записи отсутствуют");

    }
  });
}

```

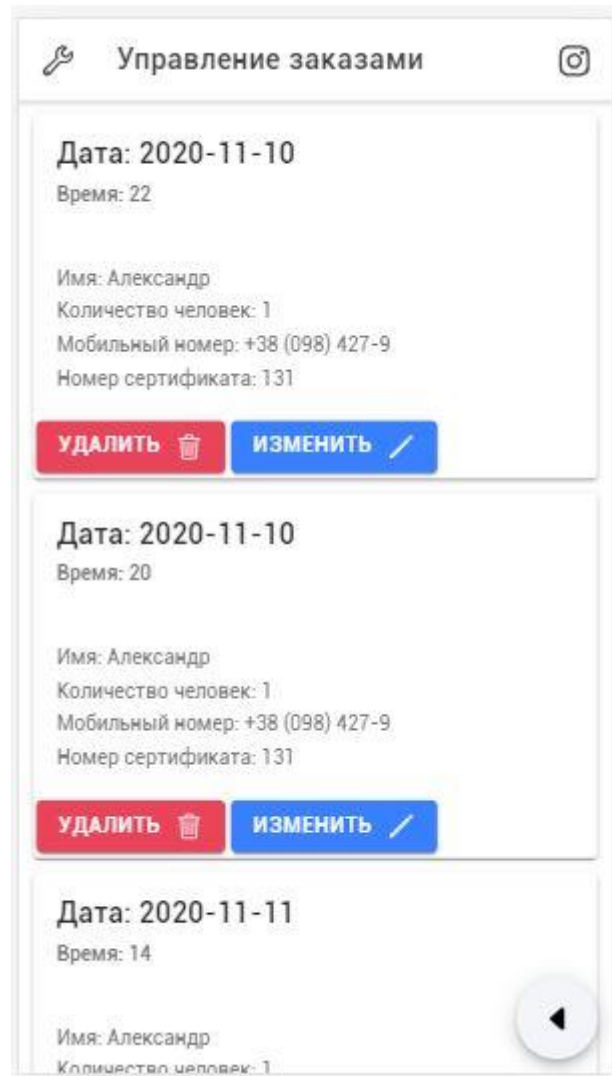


Рисунок 5.5 Результат обработки JSON данных

JSON (англ. JavaScript Object Notation, укр. запис об'єктів JavaScript) — це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дає змогу описувати об'єкти та інші структури даних. Цей формат використовується переважно для передачі

структурованої інформації через мережу (завдяки процесу, що називають серіалізацією).

Розробив і популяризував формат Дуглас Крокфорд.

JSON знайшов своє головне призначення в написанні веб-програм, а саме при використанні технології AJAX. JSON, що використовується в AJAX, виступає як заміна XML (використовується в AJAX) під час асинхронної передачі структурованої інформації між клієнтом та сервером. При цьому перевагою JSON перед XML є те, що він дозволяє складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою JavaScript в об'єкти.

5.5.1 Використання

За рахунок своєї лаконічності в порівнянні з XML, формат JSON може бути більш придатним для серіалізації складних структур.

Якщо говорити про веб-застосунки, у такому ключі він доречний у задачах обміну даними як між браузером і сервером (AJAX), так і між самими серверами (програмні HTTP-інтерфейси). Формат JSON так само добре підходить для зберігання складних динамічних структур у реляційних базах даних або файловому кеші.

5.6 Використання MySQL Бази даних

ue418458_db



Рисунок 5.6.1 Схема бази даних mysql

MySQL — вільна система керування реляційними базами даних.

MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL — компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

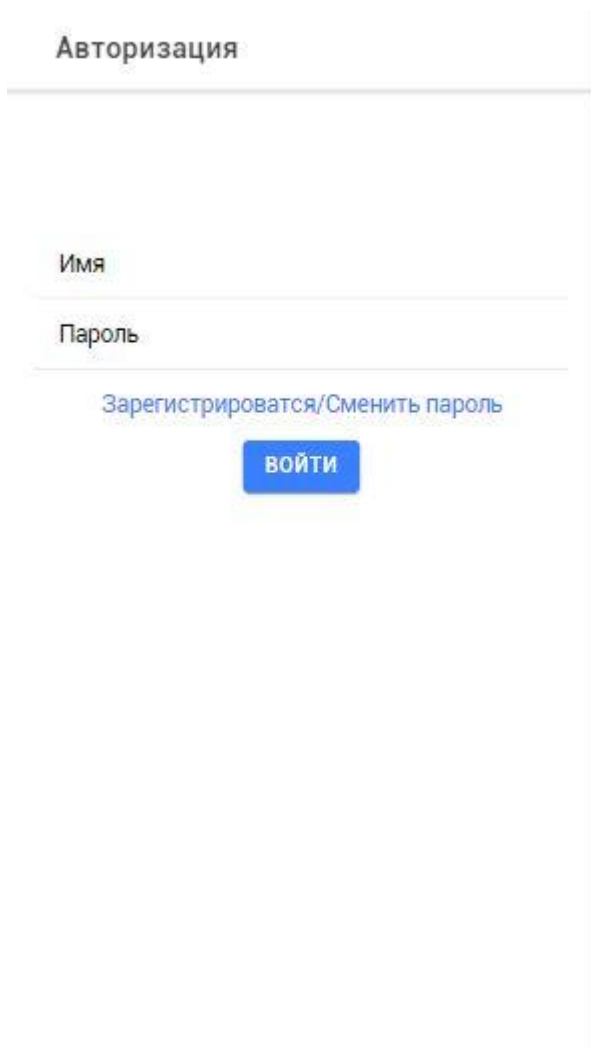
5.7 Побудова apk мобільного додатку за допомогою Cordova

Для побудови apk потрібно:

- створити проект Cordova командою `cordova create NameProject`
- перейти в папку проекту командою `cd NameProject`
- додати платформу android до проекту командою `cordova platforms add android`
- перемістити всі файли до теки `www`
- запустити будівництво командою `cordova build`
- після успішного закінчення забрати файл можна за шляхом `NameProject\platforms\android\app\build\outputs\apk\debug\app-debug.apk`

6 РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ

6.1 Екран авторизації



Авторизация

Имя

Пароль

[Зарегистрироваться/Сменить пароль](#)

ВОЙТИ

Рисунок 6.1.1 Екран авторизації

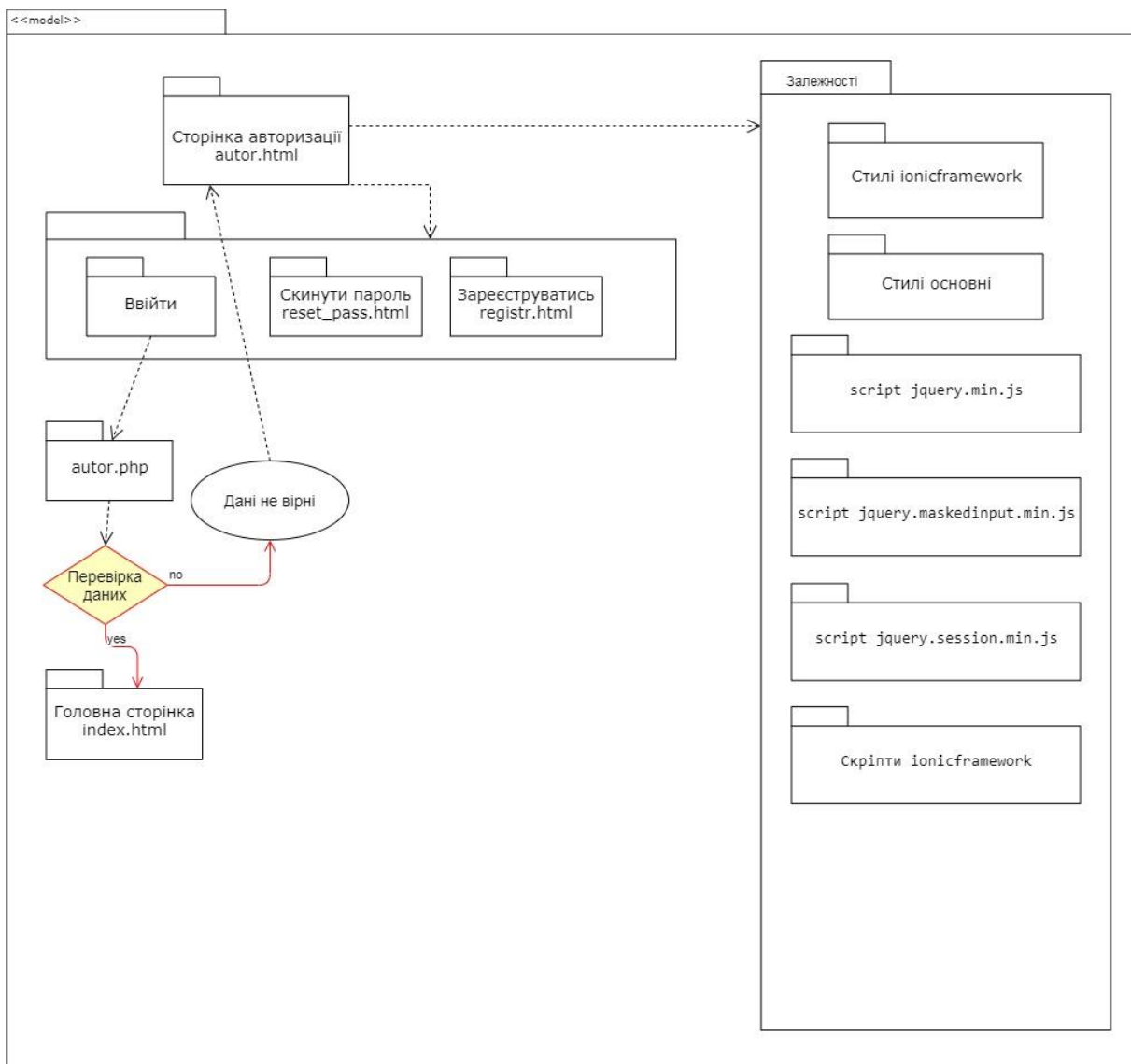
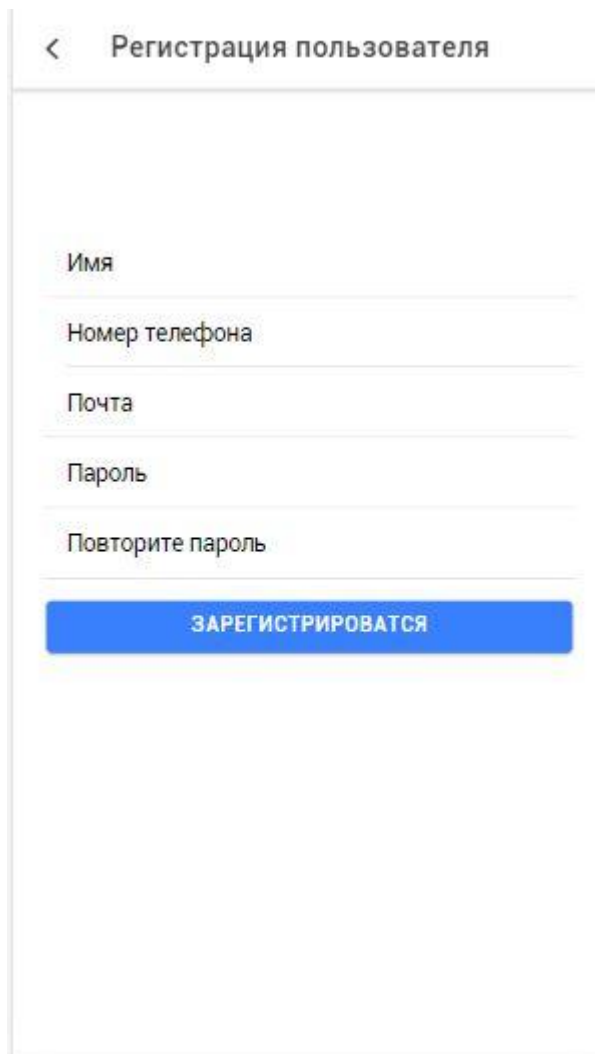


Рисунок 6.1.2 Діаграма пакетів екрану авторизації

6.2 Екран реєстрації користувачів



The image shows a mobile application screen for user registration. At the top, there is a back arrow and the title "Регистрация пользователя". Below the title are five input fields for registration details: "Имя", "Номер телефона", "Почта", "Пароль", and "Повторите пароль". At the bottom of the form is a blue button with the text "ЗАРЕГИСТРИРОВАТЬСЯ".

Рисунок 6.2.1 Екран реєстрації користувачів

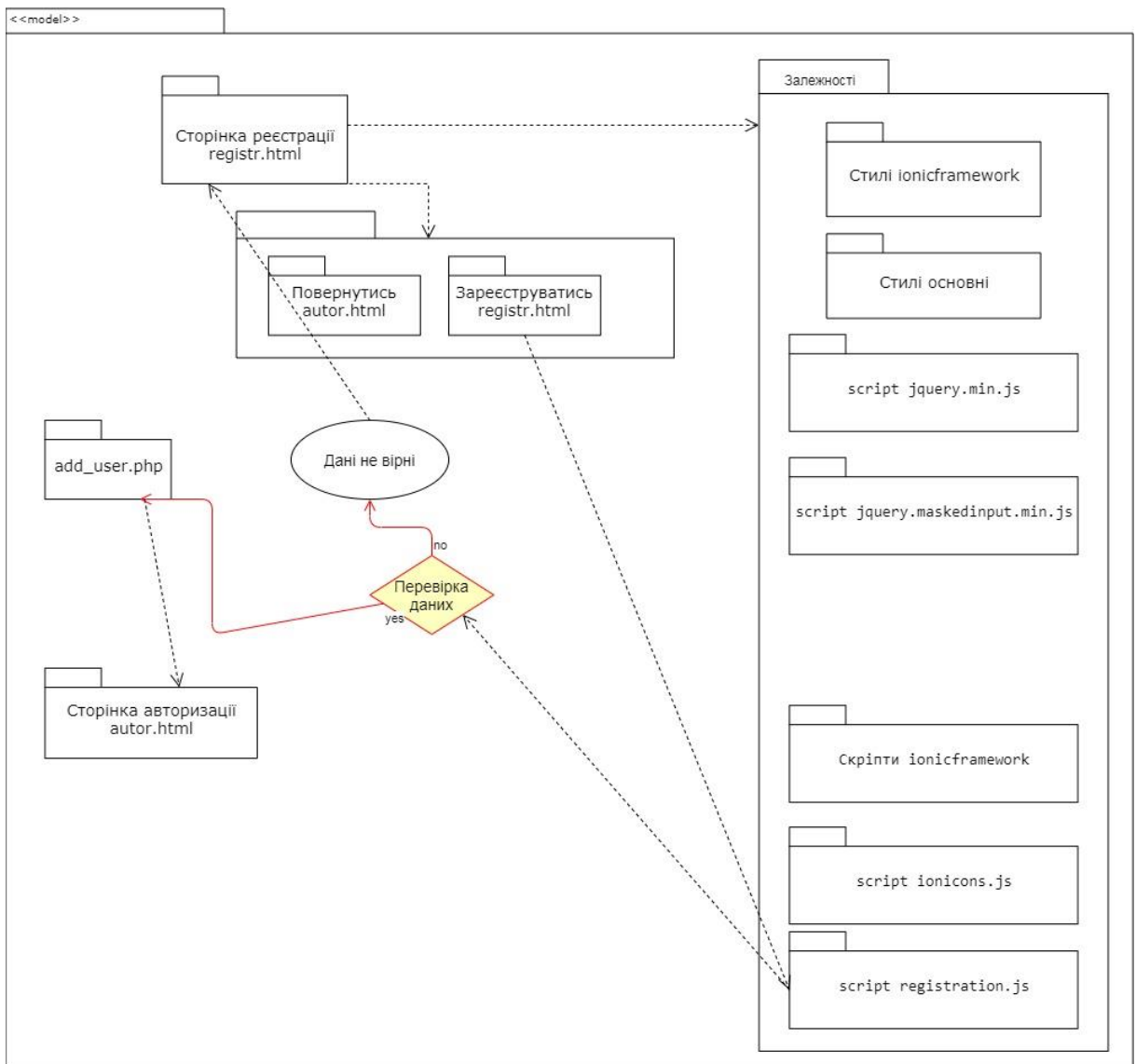
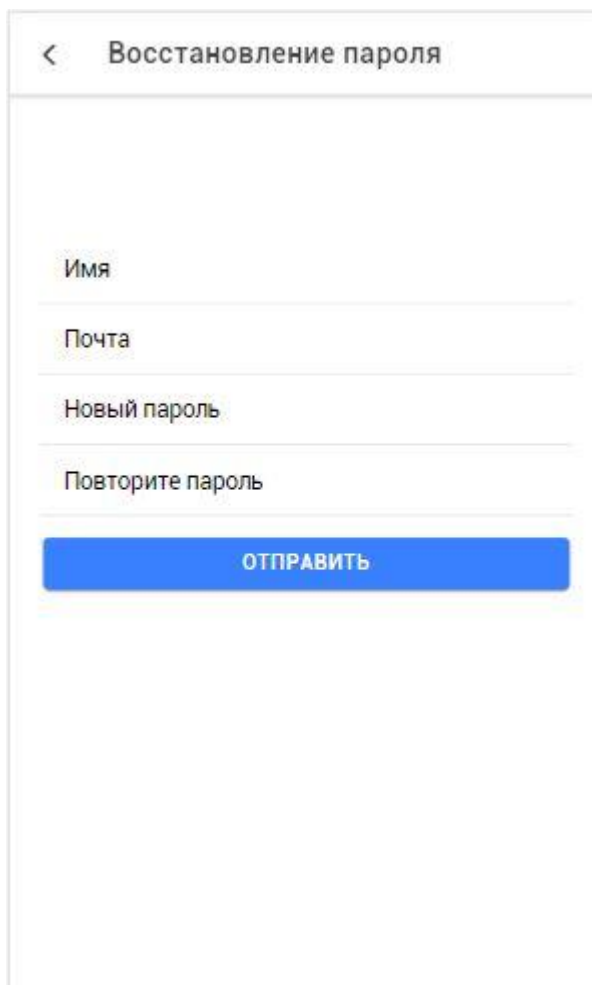


Рисунок 6.2.2 Діаграма пакетів екрану реєстрації

6.3 Екран відновлення пароля



The image shows a mobile application screen for password recovery. At the top, there is a header with a back arrow and the text "Восстановление пароля". Below the header are four text input fields: "Имя", "Почта", "Новый пароль", and "Повторите пароль". At the bottom of the form is a blue button with the text "ОТПРАВИТЬ".

Рисунок 6.3.1 Екран відновлення пароля

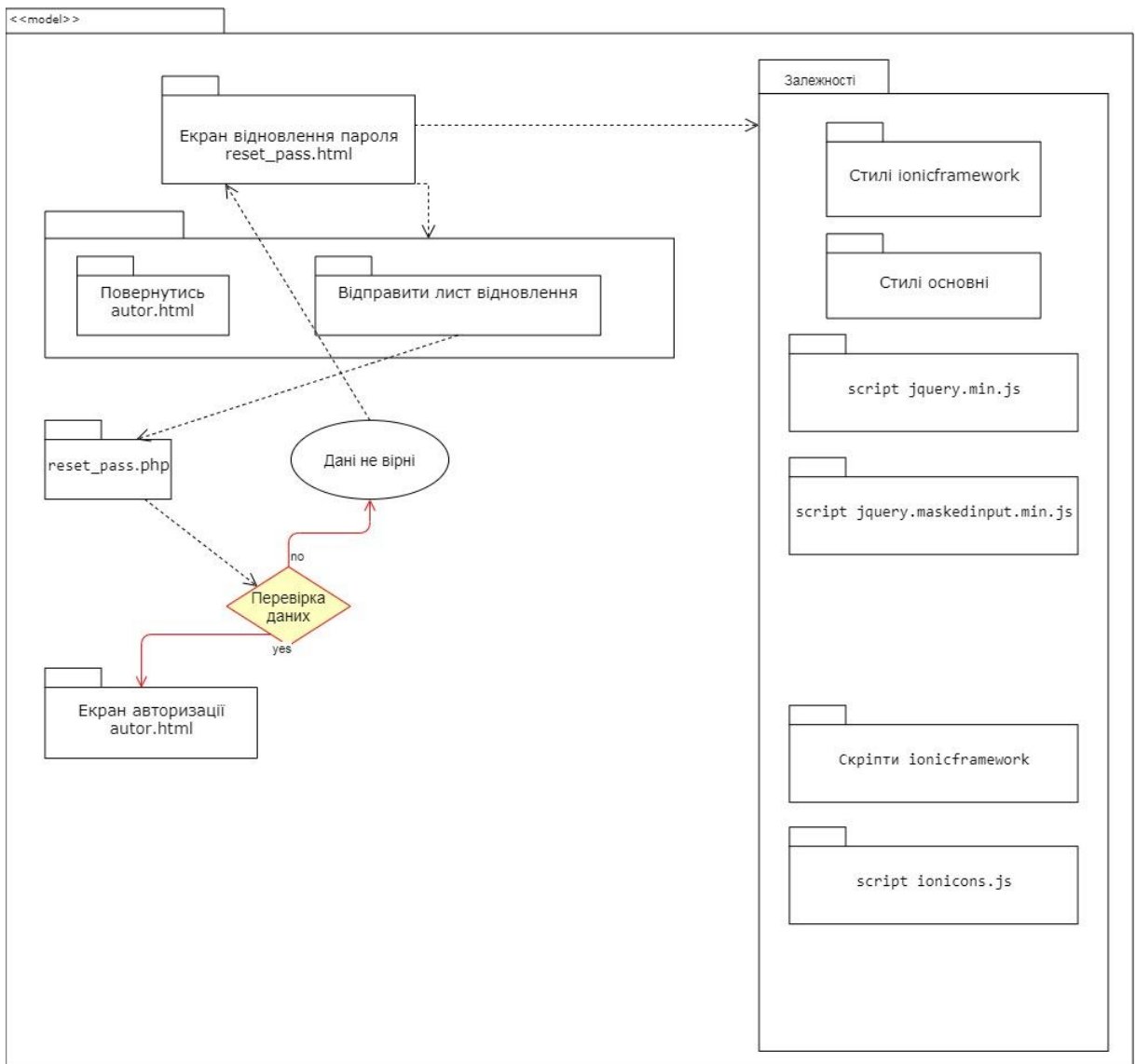


Рисунок 6.3.2 Діаграма пакетів екрану відновлення пароля

6.4 Головной экран



Рисунок 6.4.1 Головной экран



Рисунок 6.4.2 Головной экран

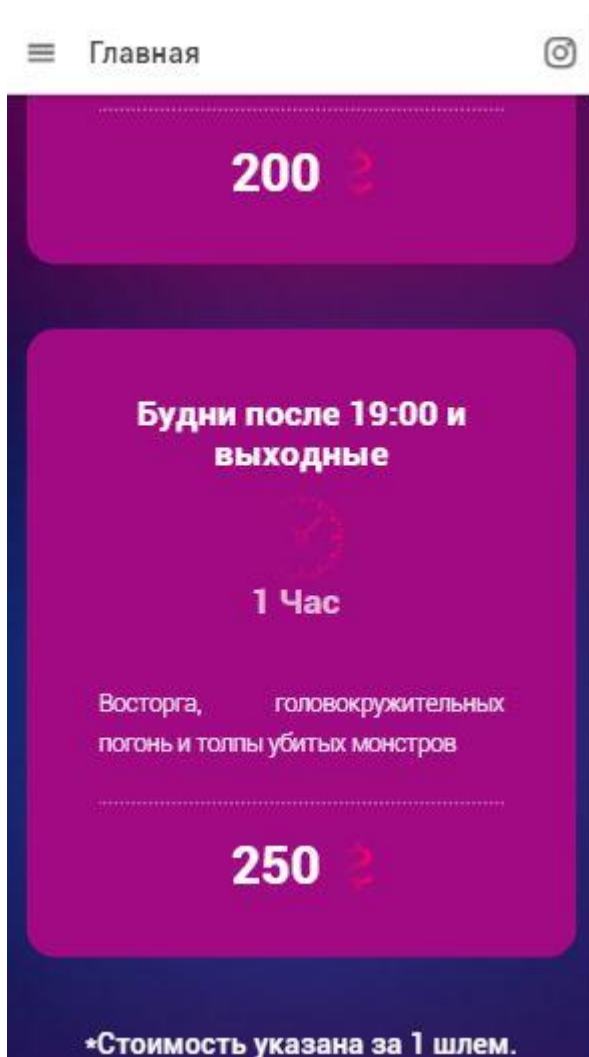


Рисунок 6.4.3 Головной экран

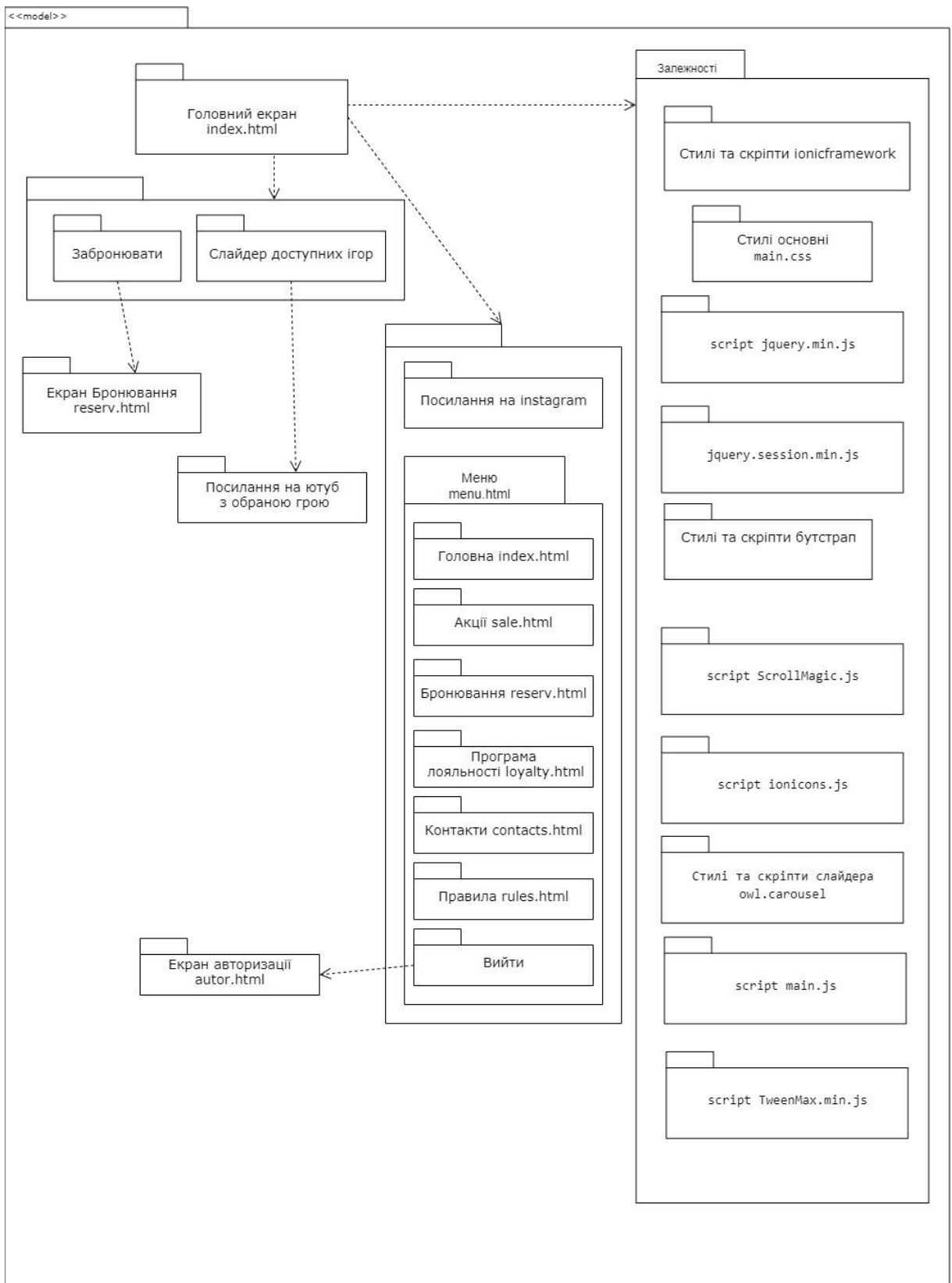


Рисунок 6.4.4 Діаграма пакетів головного екрану

6.5 Екран акцій

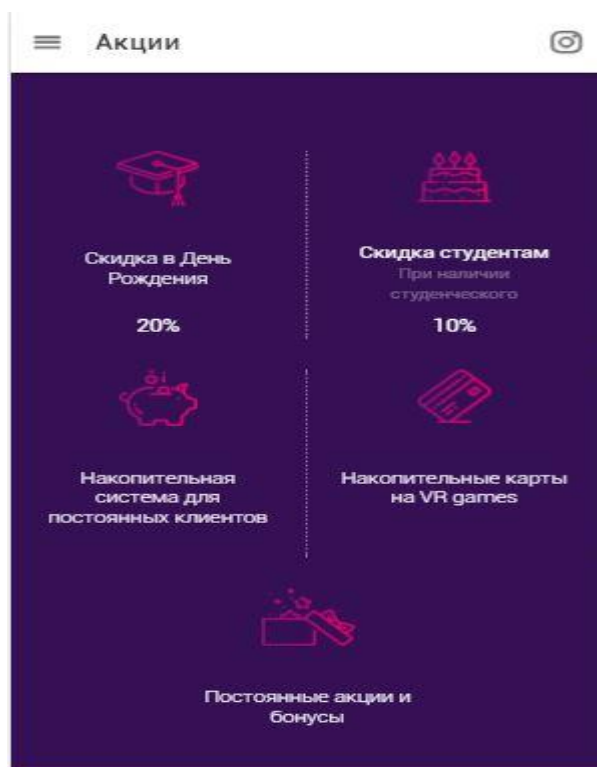


Рисунок 6.5.1 Екран акцій

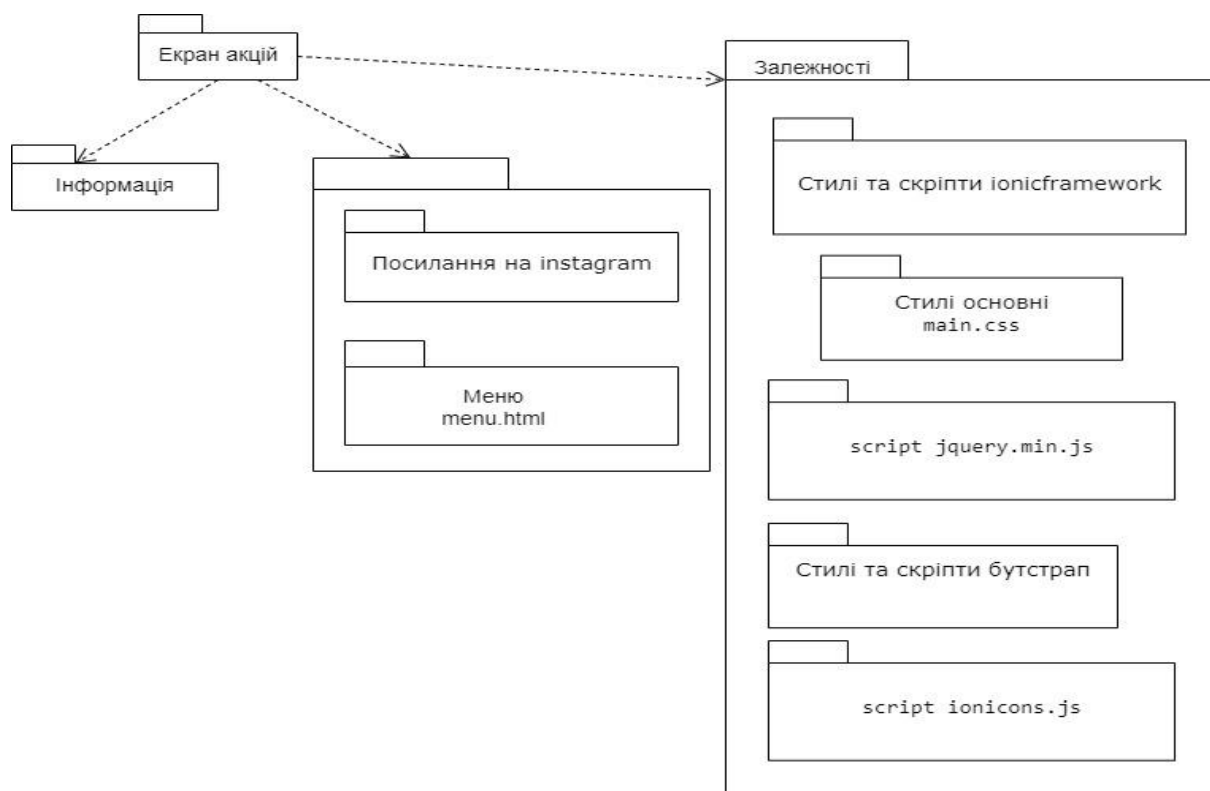


Рисунок 6.5.2 Діаграма пакетів екрану акцій

6.6 Экран онлайн бронювання

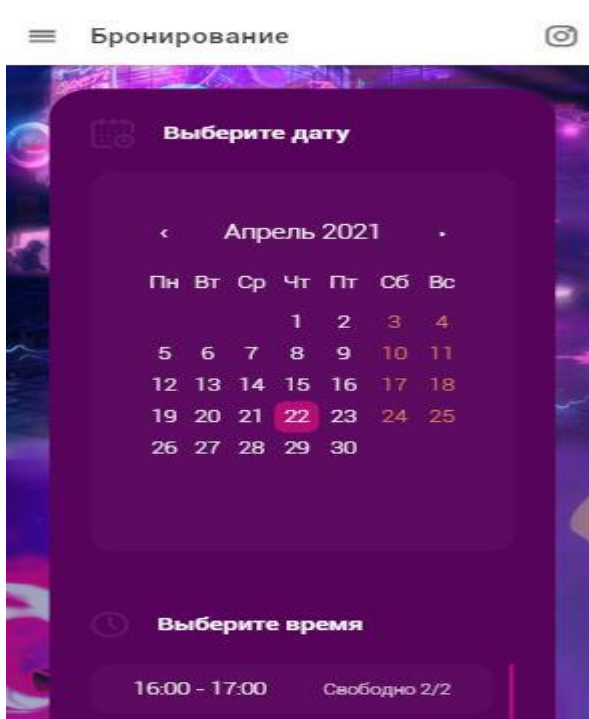


Рисунок 6.6.1 Экран онлайн бронювання

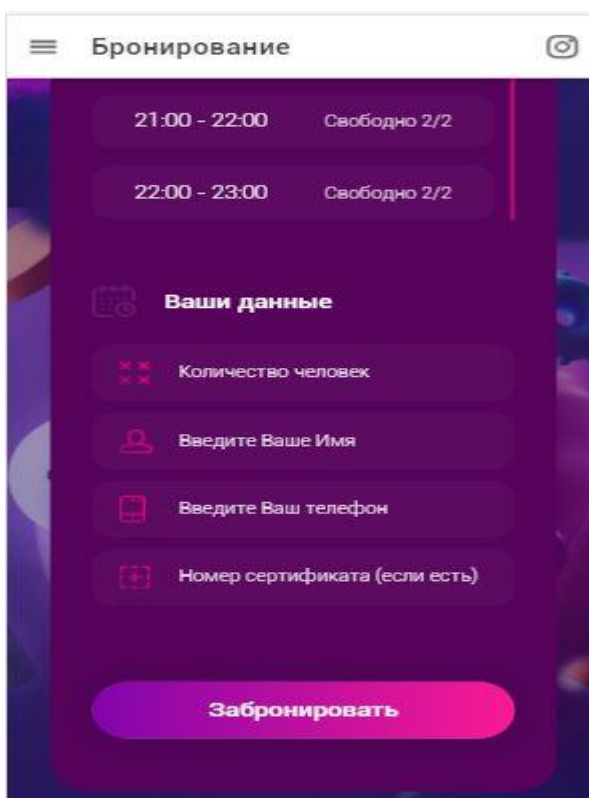


Рисунок 6.6.2 Экран онлайн бронювання

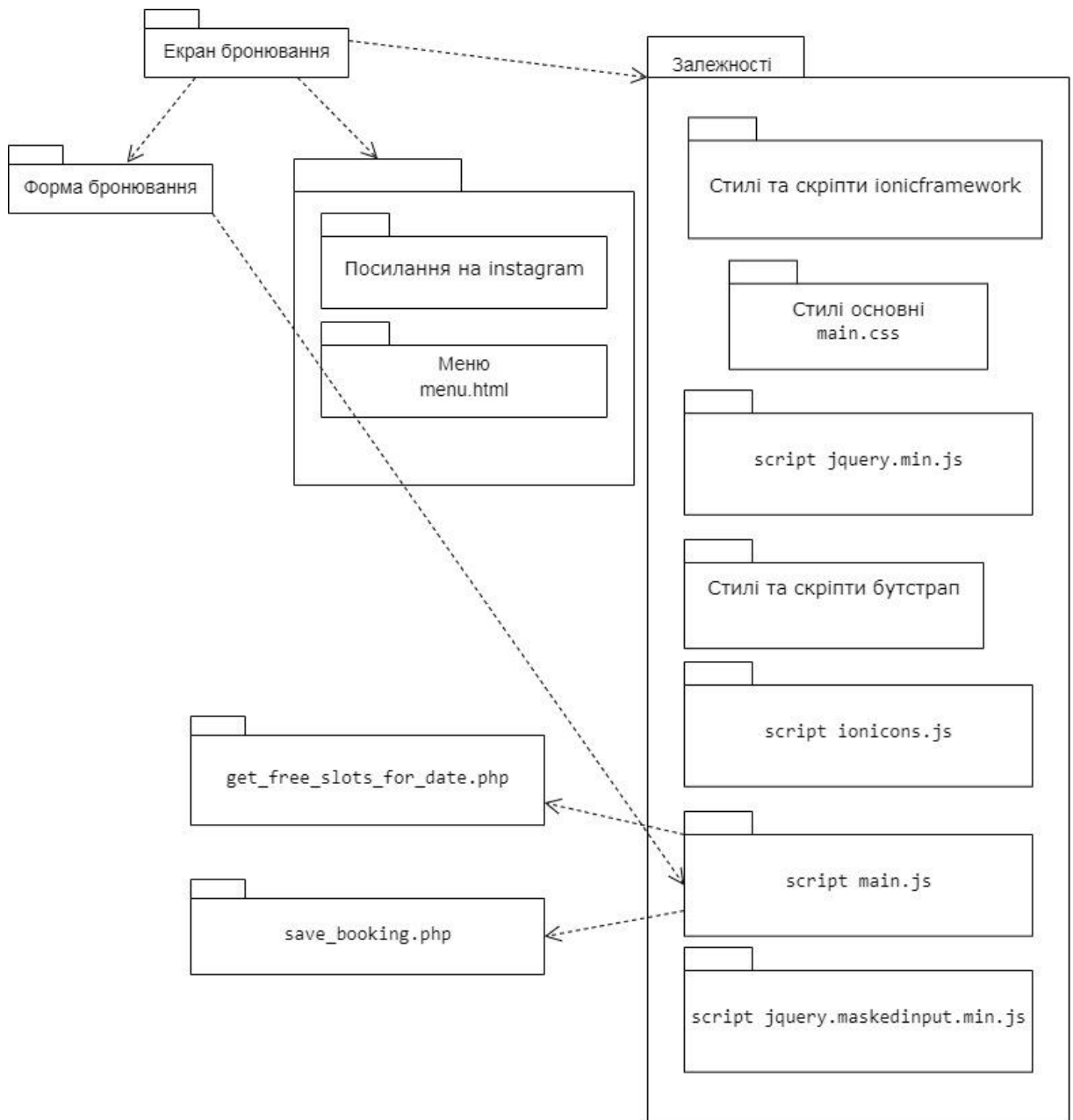


Рисунок 6.6.3 Діаграма пакетів екрану бронювання

6.7 Экран програми лояльності

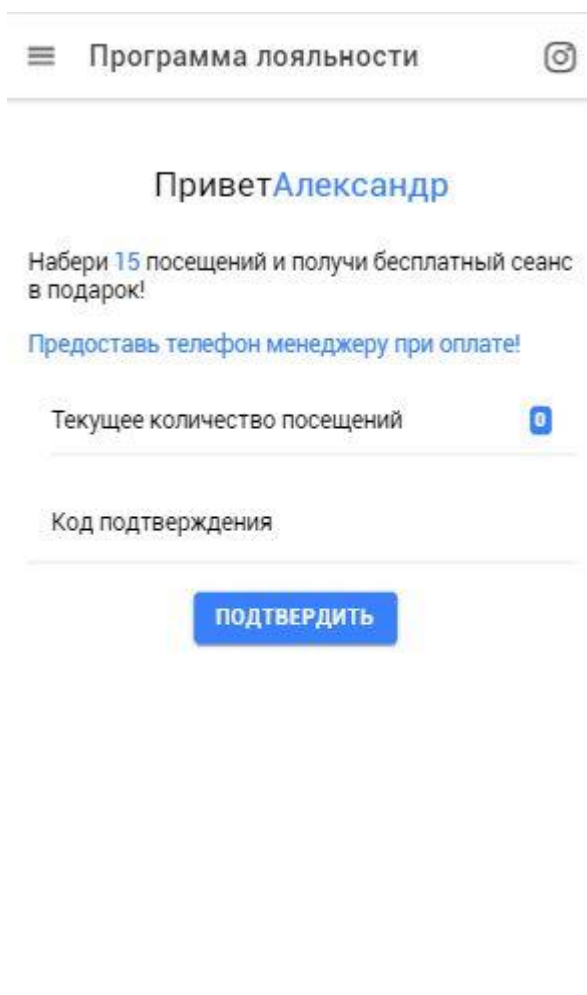


Рисунок 6.7.1 Экран програми лояльності

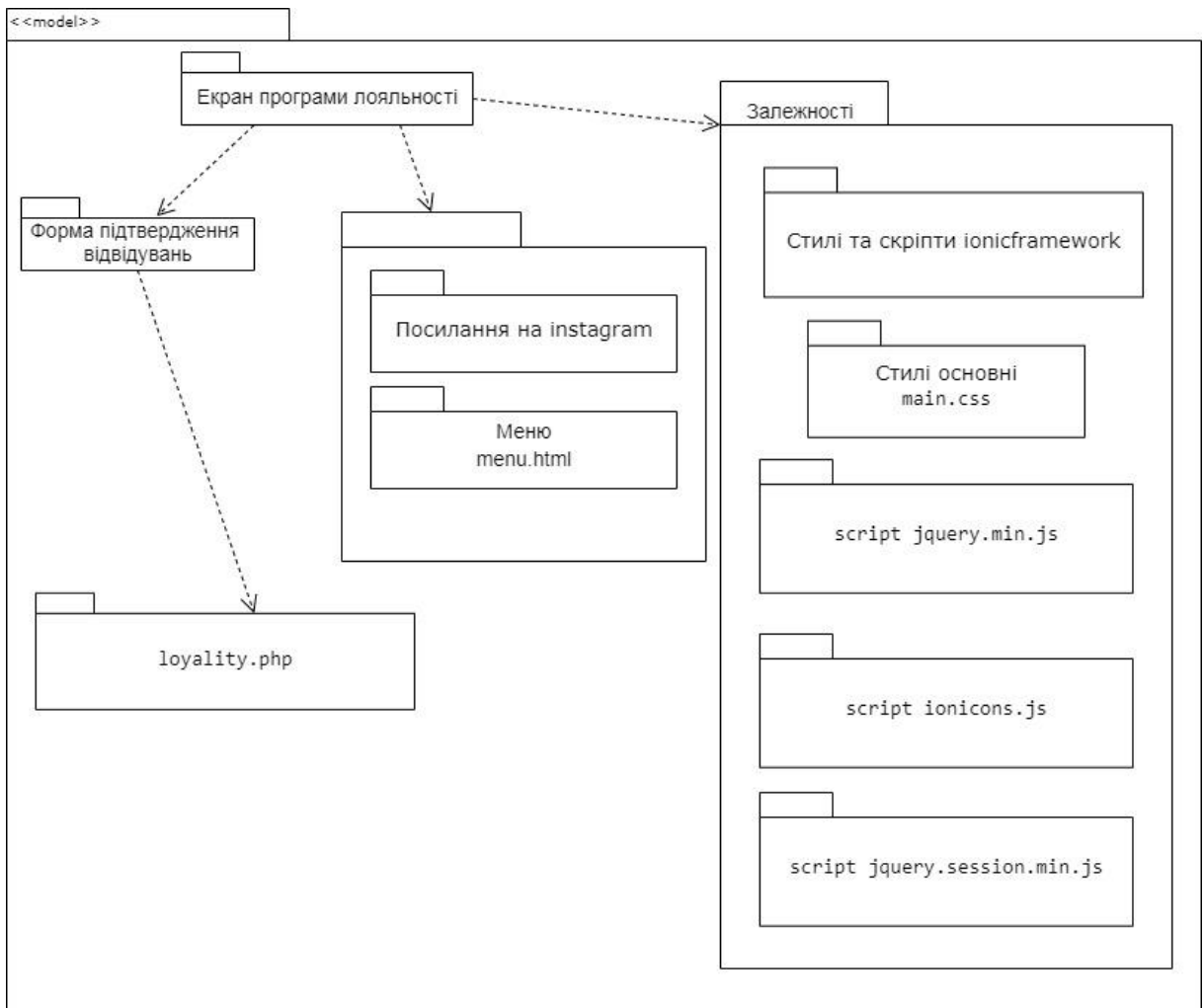


Рисунок 6.7.2 Діаграма пакетів екрану програми лояльності

6.8 Екран контактів

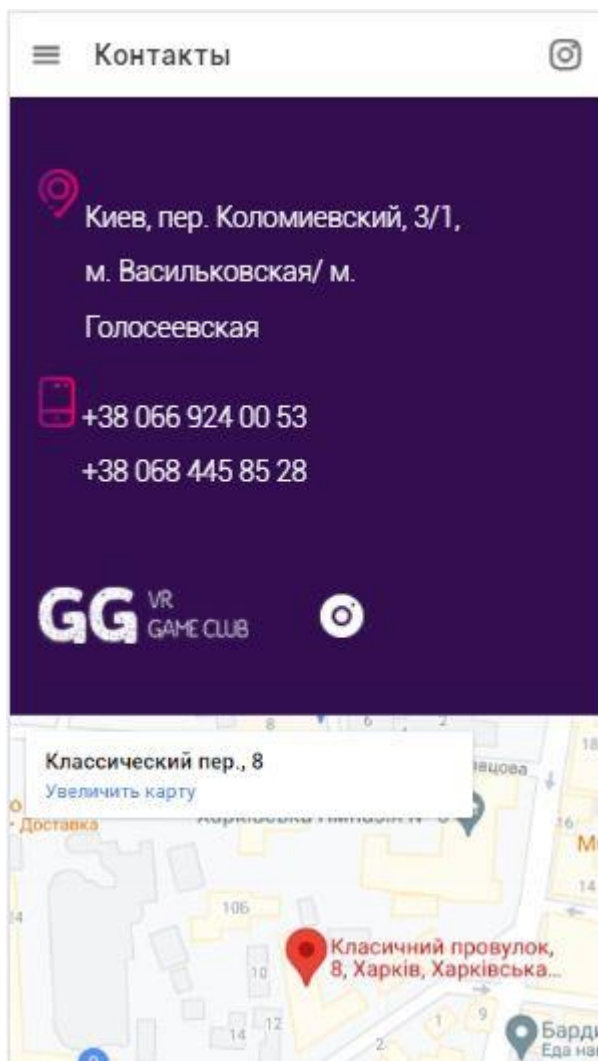


Рисунок 6.8.1 Екран контактів

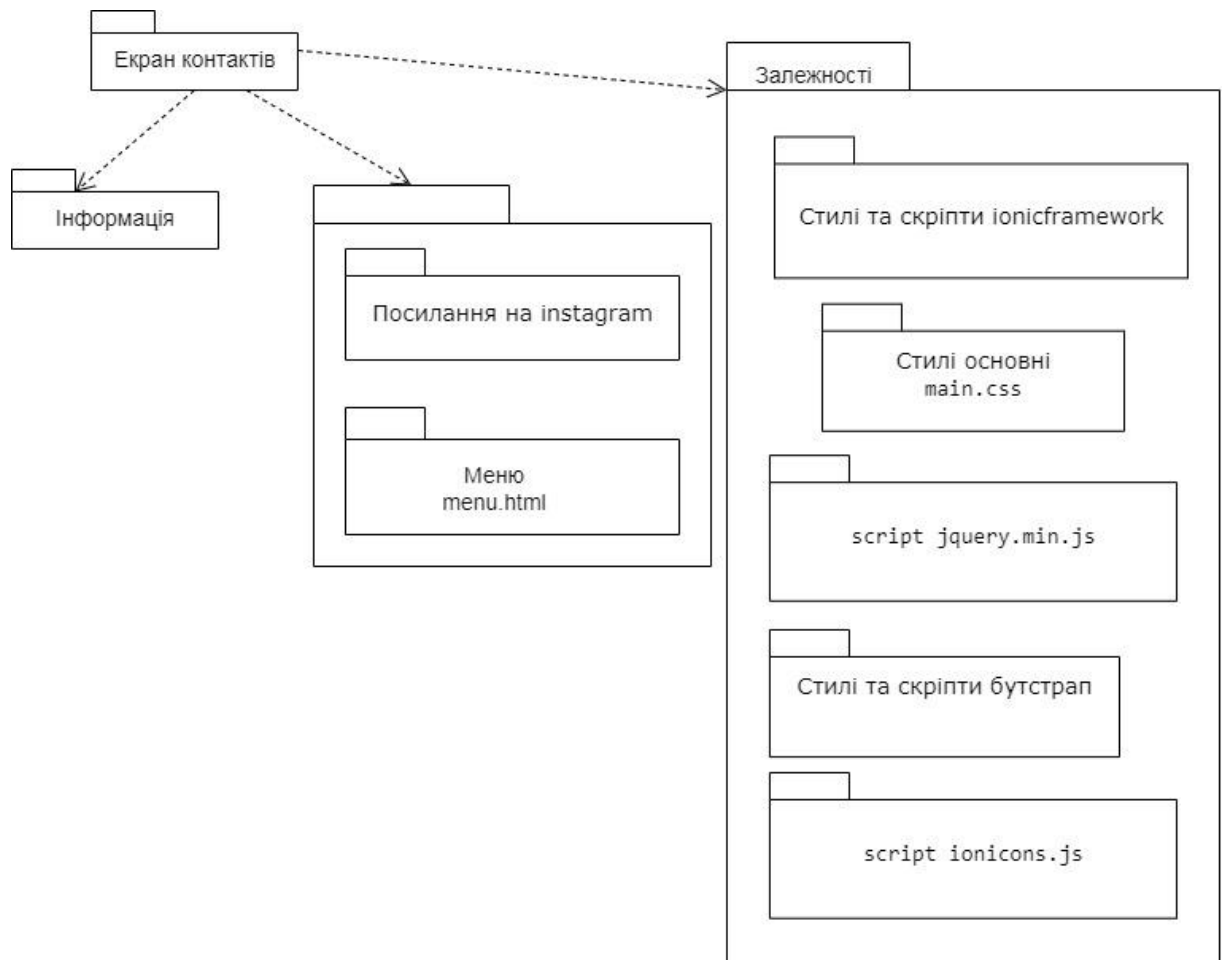


Рисунок 6.8.2 Діаграма пакетів екрану контактів

6.9 Экран правил

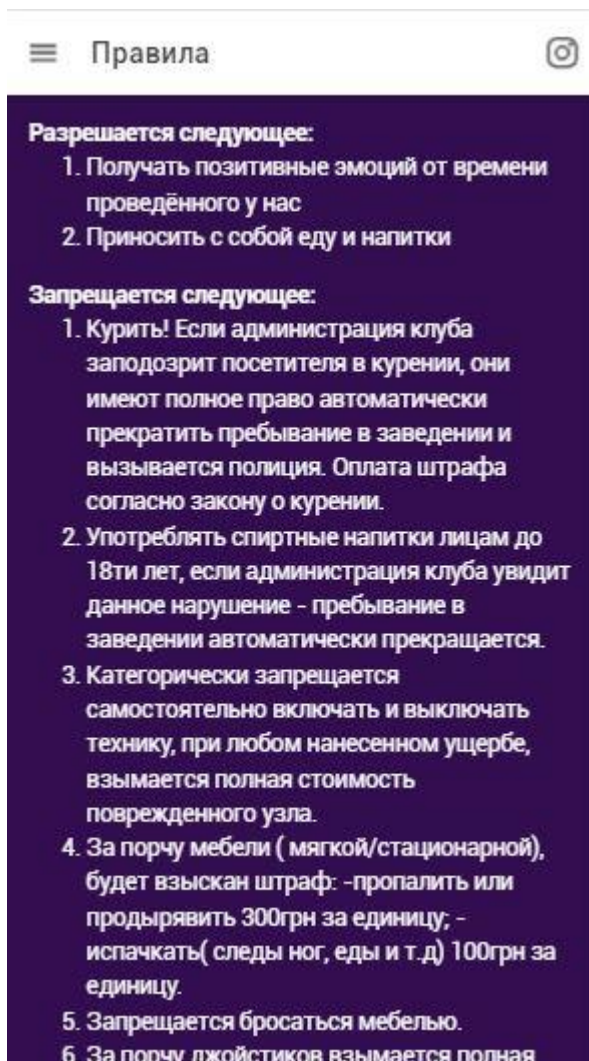


Рисунок 6.9.1 Экран правил

Діаграма пакетів екрану правил аналогічна екрану контактів.

6.10 Экран керування замовленнями

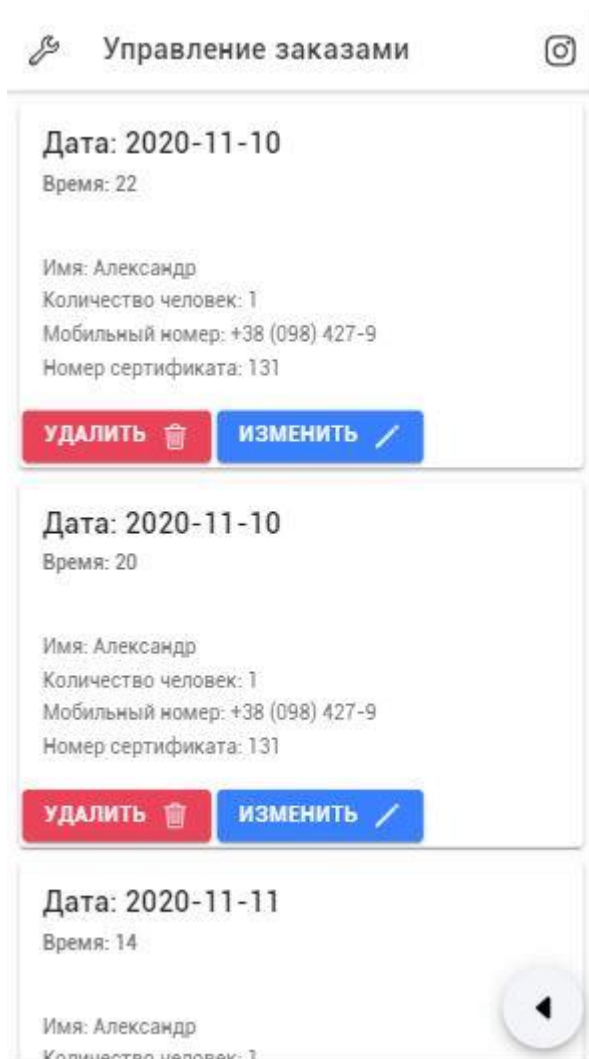


Рисунок 6.10.1 Экран керування замовленнями

🔧 Управление заказами 📷

Дата: 2020-11-10
Время: 20
Имя: Александр
Количество человек: 1
Мобильный номер: +38 (098) 427-9
Номер сертификата: 131

НАЗАД ↩️ ПОДТВЕРДИТЬ ✓

Дата: 2020-11-11
Время: 14
Имя: Александр
Количество человек: 1
Мобильный номер: +38 (098) 427-9
Номер сертификата: 131

УДАЛИТЬ 🗑️ ИЗМЕНИТЬ ✎

◀️

Рисунок 6.10.2 Экран керування замовленнями



Рисунок 6.10.3 Меню адміністратора екрану керування замовленнями

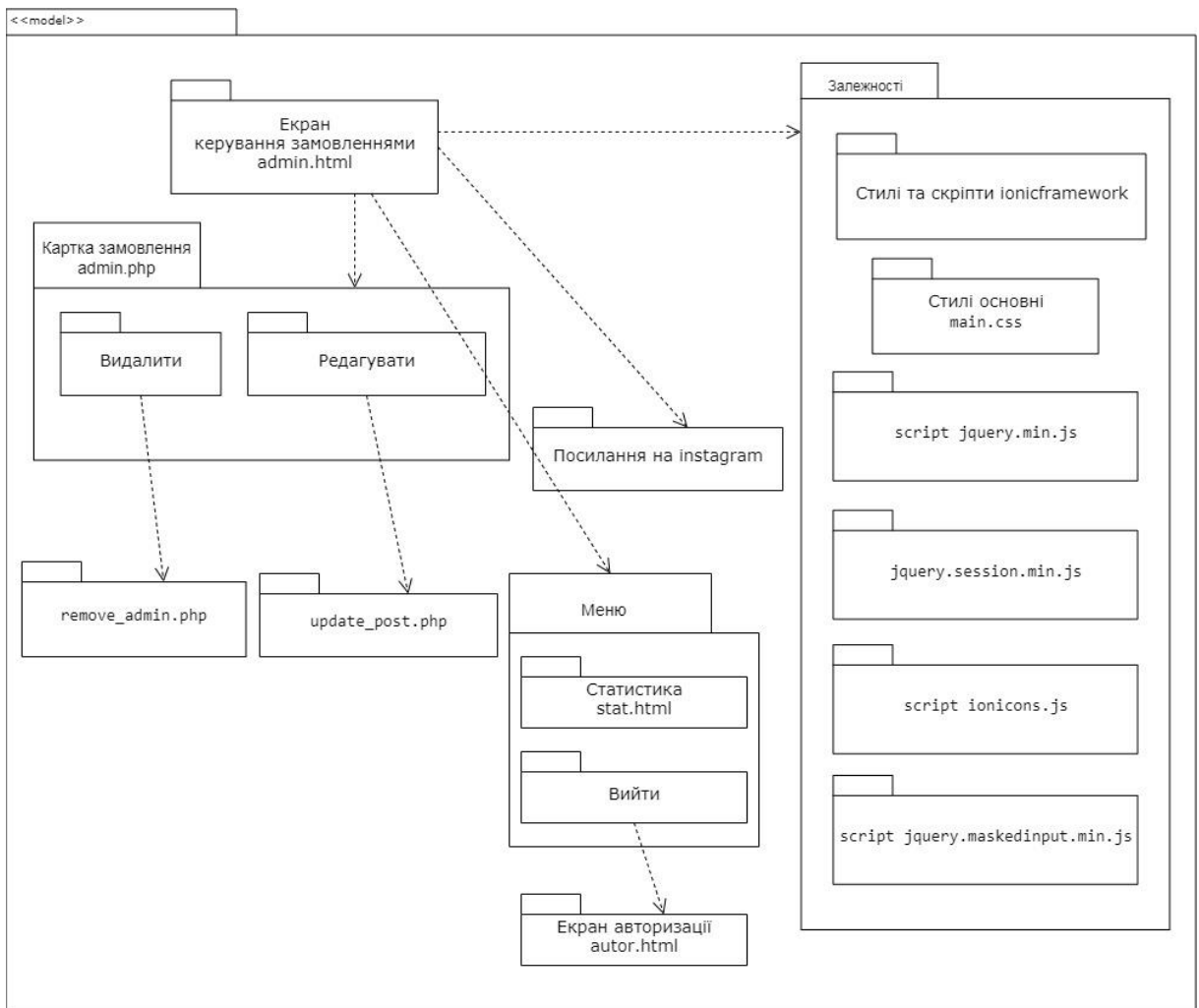


Рисунок 6.10.4 Діаграма пакетів екрану керування замовленнями

6.11 Екран статистики

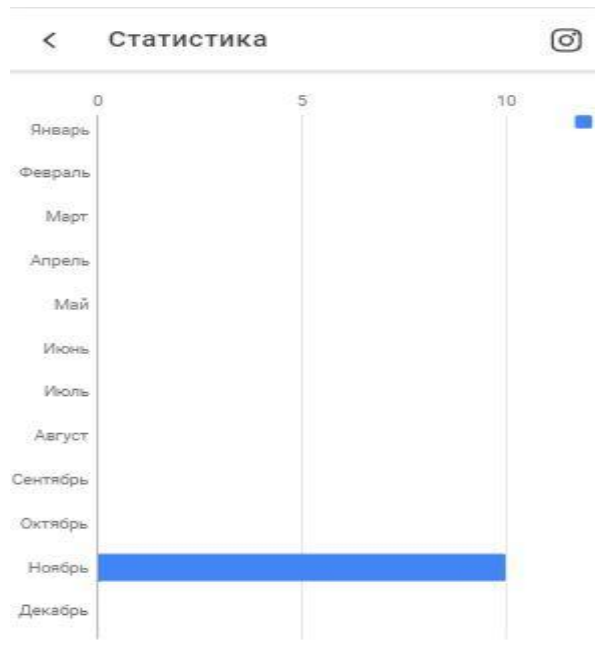


Рисунок 6.11.1 Екран статистики

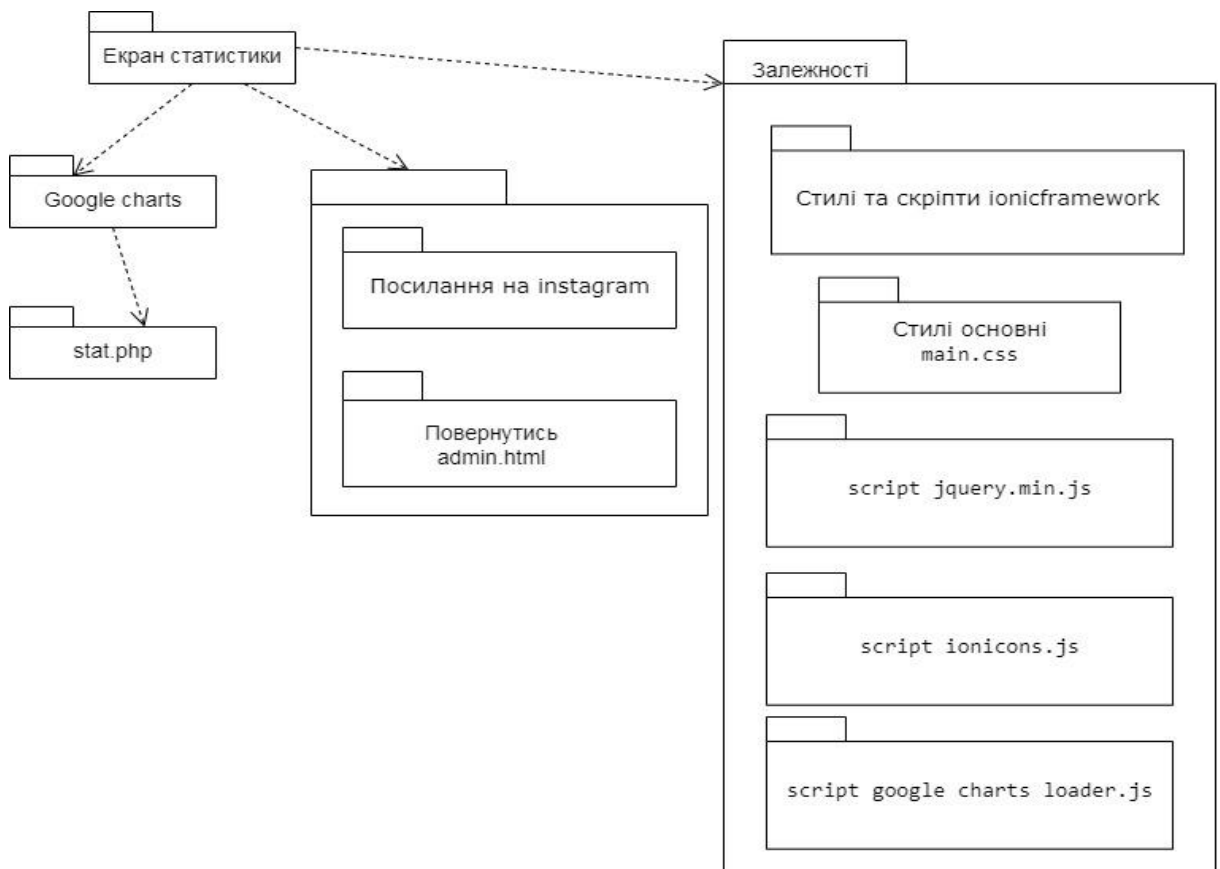


Рисунок 6.11.2 Діаграма пакетів екрану статистики

6.12 Діаграма предметної галузі

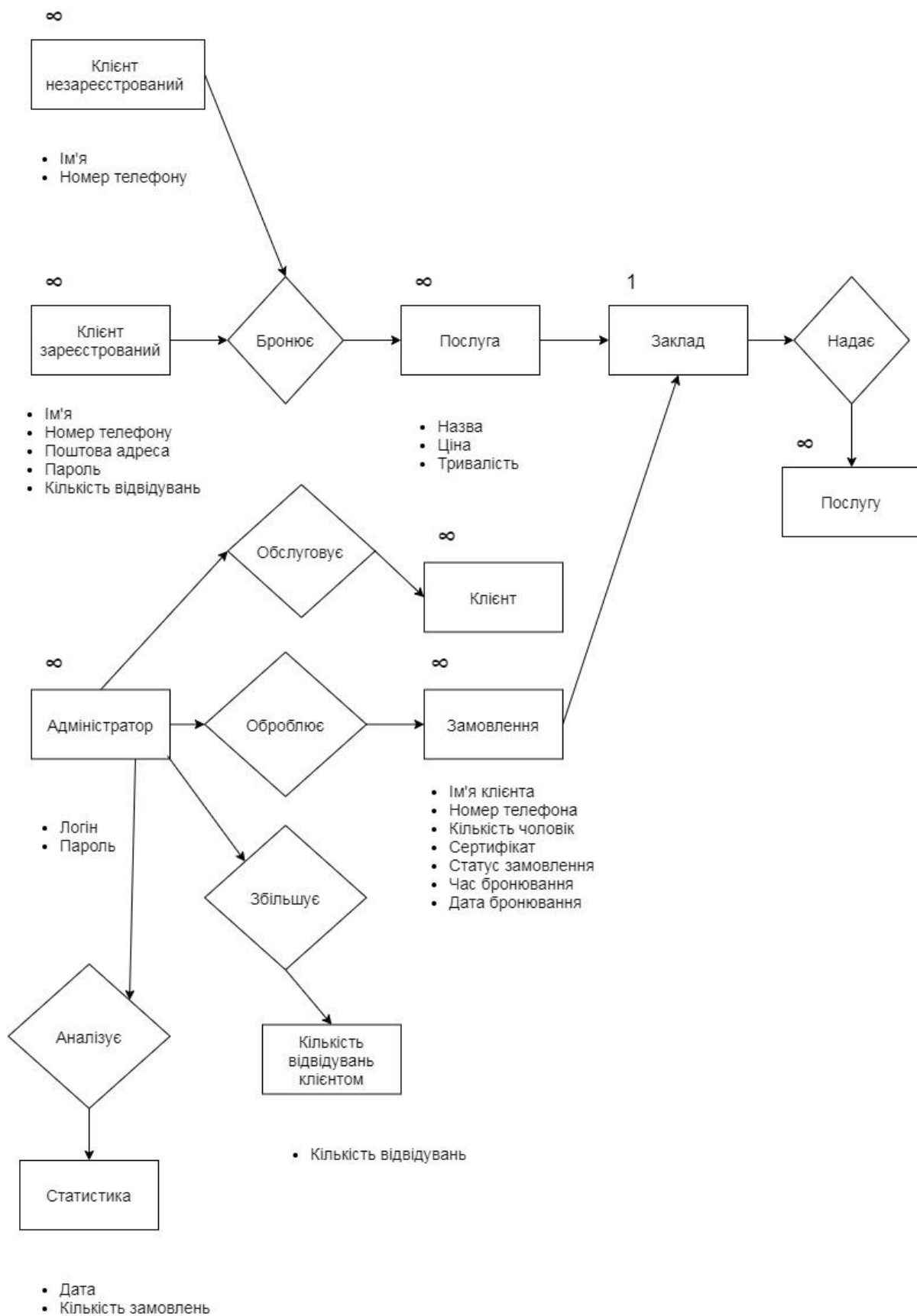


Рисунок 6.12.1 Діаграма предметної галузі

6.13 Діаграма артефактів

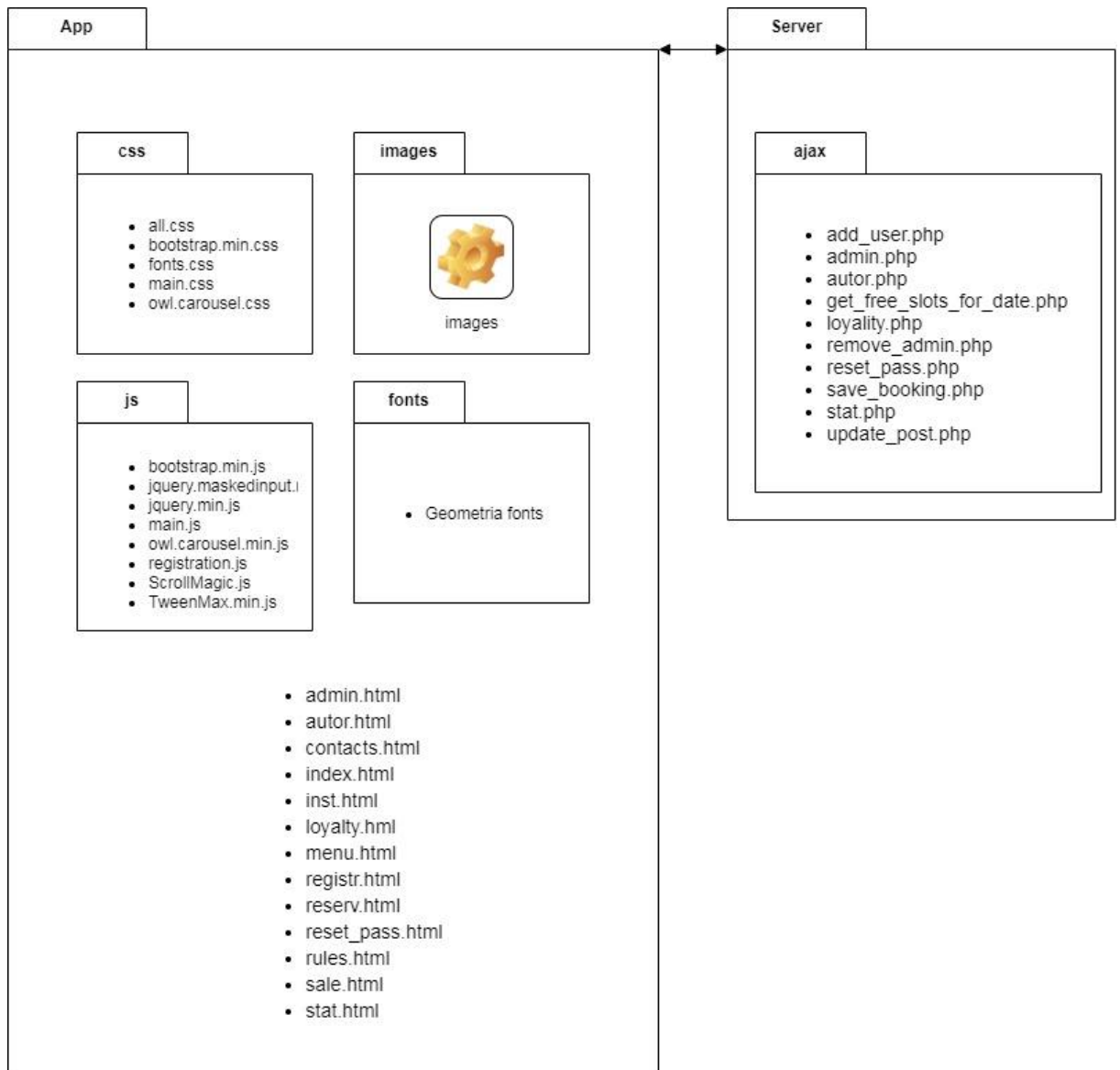


Рисунок 6.13.1 Схема структури програми

7 РОЗГОРТАННЯ

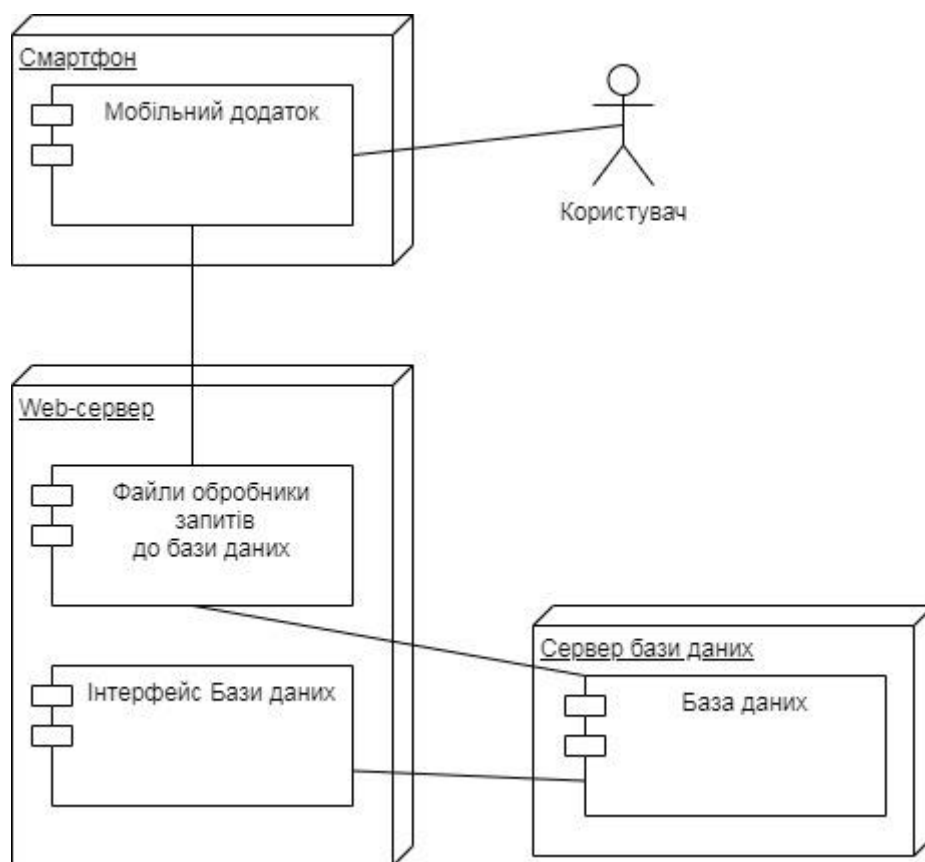


Рисунок 7.1 Діаграма розгортання

Діаграма розгортання (англ. *deployment diagram*) — діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

ВИСНОВКИ

У ході роботи було розроблено мобільний додаток для бронювання місць в закладах, що надають послуги віртуальної реальності мовою javascript.

Для цього виконано наступні дії:

1. Проаналізовано існуючі додатки на ринку та методи програмного моделювання, направлені для вирішення подібних задач. Даний аналіз став підґрунтям у виборі методик та платформ для проектування та реалізації даної системи.

2. Обрана існуюча модель, яка найбільше відповідає поставленим задачам по розробці програмного забезпечення.

3. Описано проектування та розробку мобільного додатка, технічних засобів (Hardware), платформ та інструментів по розробці програмного забезпечення (Software).

4. Зображено процес розробки гібридного мобільного додатка.

5. Розроблено мобільний додаток для бронювання місць в закладах, що надають послуги віртуальної реальності мовою javascript на базі операційної системи Android.

Розроблена система дасть змогу створювати мобільні додатки без знань фреймворків мов програмування а також вона дозволяє зменшити витрати часу на оформлення зовнішнього вигляду програми оскільки ionicframework задовільнить більшість розробників та користувачів своєю базою готових рішень.

Дана система може бути модефікована для інших проектів. Тому в майбутньому планується модернізувати дану систему для можливості оплати замовлення онлайн.

Висновок до пункту 5.1

Ionic - відмінна технологія, що дозволяє зробити готове для випуску додаток набагато швидше, ніж при традиційній розробці. Великий плюс цього підходу - те, що замість Java / Swift / Objective-C можна використовувати стандартні технології веб-розробки, такі як HTML, CSS, JavaScript / TypeScript і

Angular. Є невеликий мінус в продуктивності - нативное додаток може бути більш чуйним на старих пристроях, але на будь-якому сучасному смартфоні різниця буде незначною.

Висновок до пункту 5.3

Якщо підсумувати сказане вище, можна дійти висновку, що jQuery – якісний спрощувач JavaScript'а, особливо для тих, хто не має бажання з ним довго мучитись. jQuery доступний, кросбраузерний, що доволі важливо при роботі. Тому рекоментую використовувати бібліотеку в своїх проектах.

ПЕРЕЛІК ПОСИЛАНЬ

1. Документація з jquery [Електронний ресурс] – Режим доступу:
<https://api.jquery.com/>
2. James Rumbaugh, Ivar Jacobson, Grady Booch (1999). The unified modeling language reference manual (англ.). Addison Wesley Longman Inc. ISBN 0-201-30998-X. [Електронний ресурс] – Режим доступу:
<https://archive.org/details/unifiedmodeling100rumb/page/n9/mode/2up>
3. Документація з javascript [Електронний ресурс] – Режим доступу:
<https://learn.javascript.ru/>
4. Форум [Електронний ресурс] – Режим доступу: <https://stackoverflow.com/>
5. Документація з PHP [Електронний ресурс] – Режим доступу:
<https://www.php.net/manual/ru/index.php>
6. Документація з ionicframework [Електронний ресурс] – Режим доступу:
<https://ionicframework.com/docs>

ДОДАТОК

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ



Навчально-науковий інститут заочного та дистанційного навчання

Кафедра Інженерії програмного забезпечення



Тема: «РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ БРОНЮВАННЯ МІСЦЬ В ЗАКЛАДАХ, ЩО НАДАЮТЬ ПОСЛУГИ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ, МОВОЮ JAVASCRIPT»

Виконав: студент 5 курсу, групи ППЗ-52
Коваль О.С.
Керівник роботи: старший викладач
кафедри
Гаманюк І.М.

Київ – 2021

Об'єкт, Предмет та Мета Дослідження

Мета роботи – покращення методики надання послуг користувачам шляхом використання розробленого програмного забезпечення.

Об'єкт дослідження – процес надання послуг користувачам

Предмет дослідження – є автоматизація процесу бронювання місць шляхом створення відповідного програмного забезпечення.

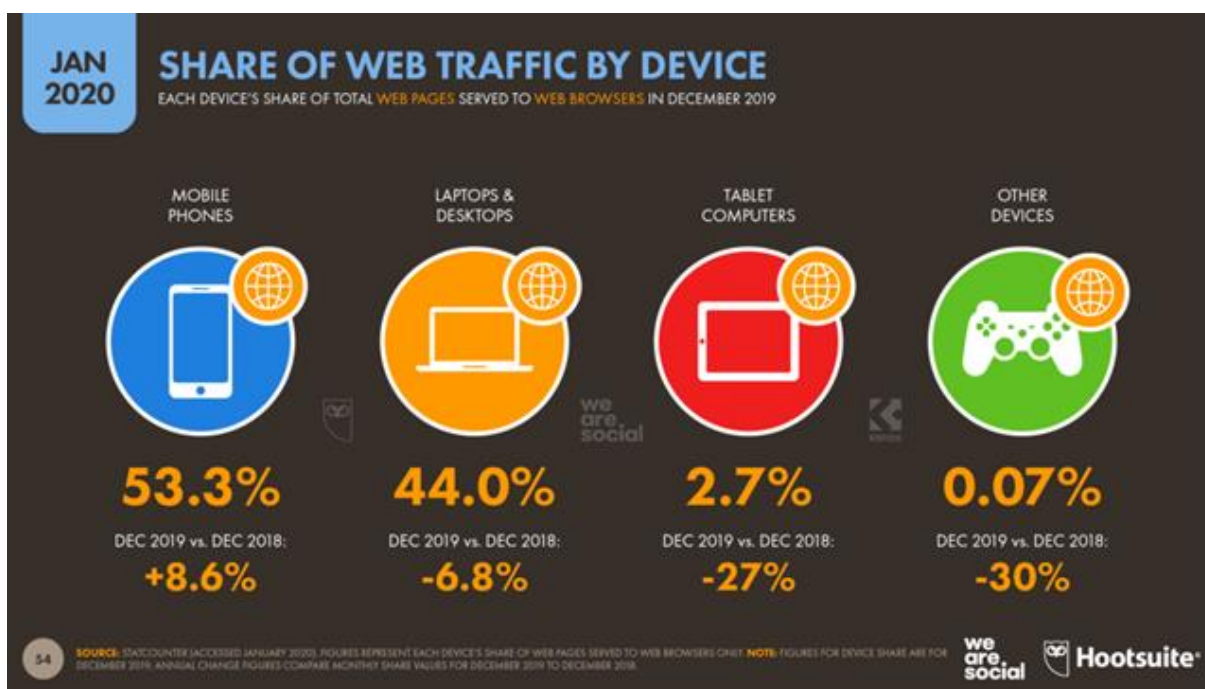
Методи дослідження – У науковій роботі було використані загальнонаукові та спеціальні методи дослідження, а саме: методи аналізу та синтезу, індукції та порівняння під час аналізу вимог та моделюванню програмного забезпечення, також графічні методи – для представлення результатів дослідження у вигляді схем та рисунків тощо.

Актуальність

За звітом Datareportal за 2020 рік

- Кількість людей, що користуються **Інтернетом** у всьому світі, зросла до **4,54 мільярда**, що на 7 відсотків більше (298 мільйонів нових користувачів) порівняно з січнем 2019 року.
- У січні 2020 року в **соціальних мережах** налічується **3,80 мільярда користувачів**, причому ця кількість зросла більш ніж на 9 відсотків (321 мільйон нових користувачів) з цього часу минулого року.
- Зараз у всьому світі понад **5,19 мільярда людей** користуються **мобільними телефонами**, а кількість користувачів за останній рік зросла на 124 мільйони (2,4 відсотка).

3



Вимоги

Адміністратори:

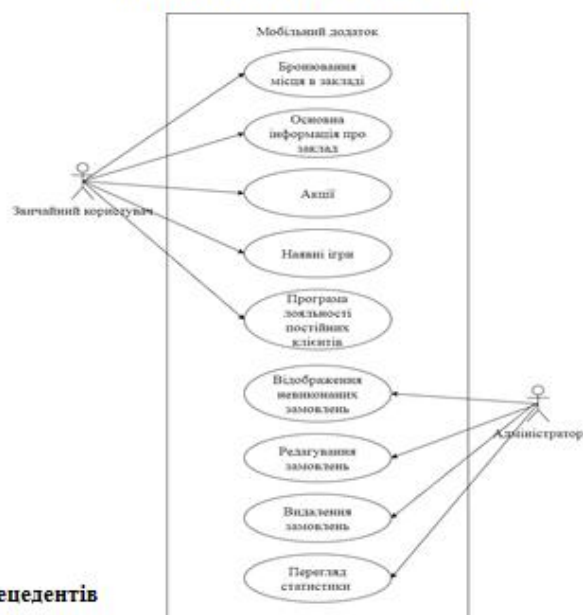
- Можливість прибирати замовлення
- Можливість змінювати замовлення
- Показ нових замовлень
- Показ статистики замовлень за місяць

Користувачі:

- Можливість забронювати місце
- Показ основної інформації про заклад:
 - 1) Геолокація
 - 2) Контакти
 - 3) Нові ігри
 - 4) Акції
 - 5) Ціни на послуги
- Програма лояльності (знижки за кількістю відвідувань)

5

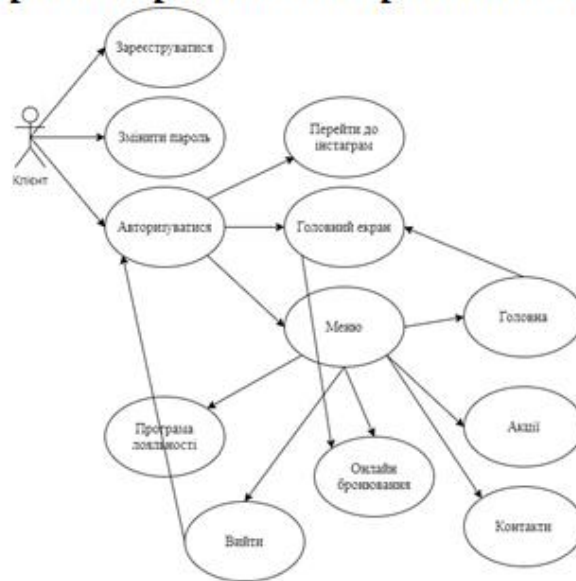
Можливості системи



Діаграма прецедентів

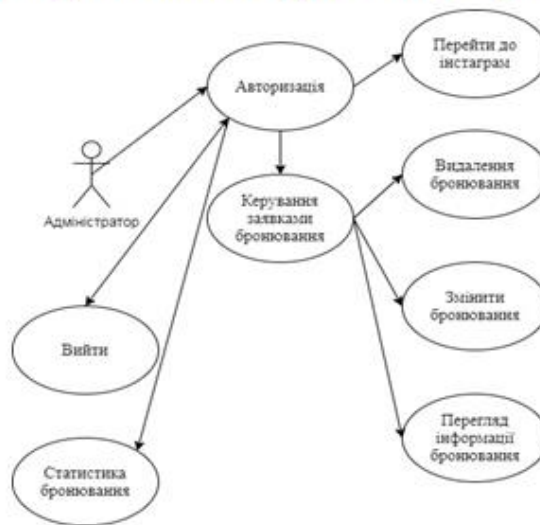
6

Діаграма варіантів використання Клієнт



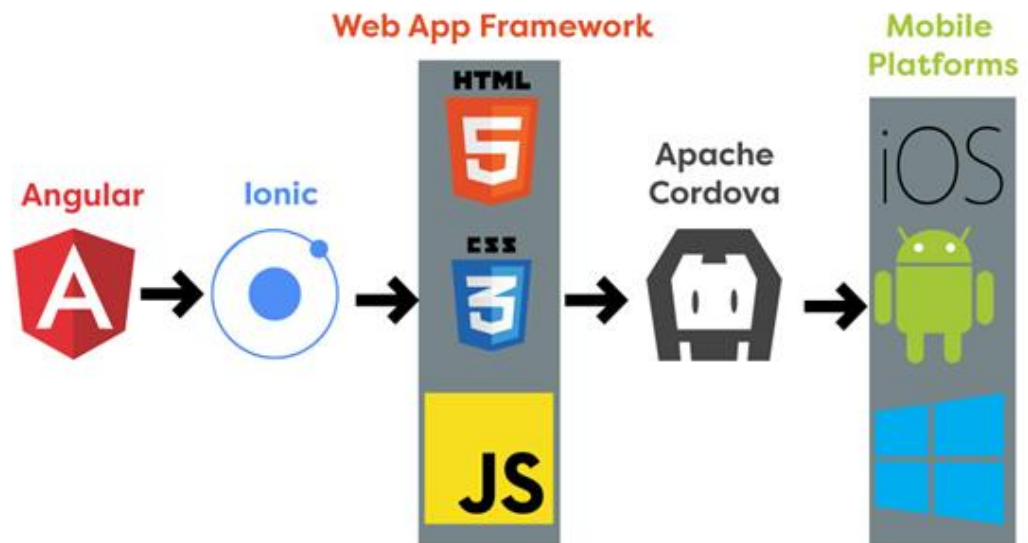
8

Діаграма варіантів використання Адміністратор



1

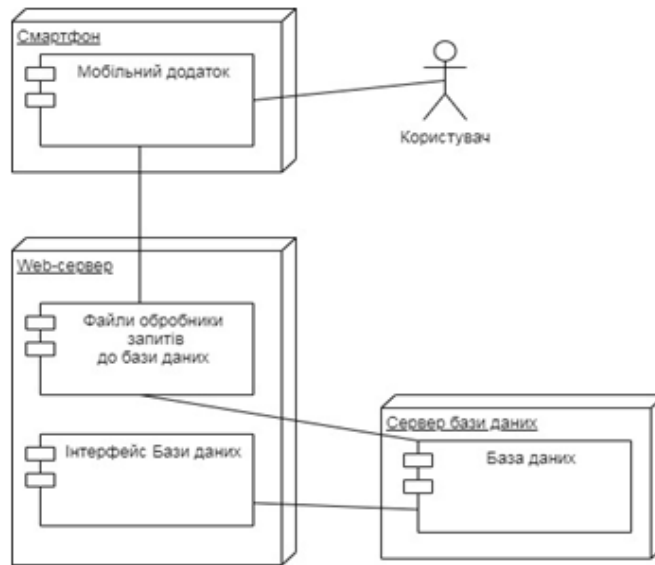
Схема побудови гібридного мобільного додатку яку я використав



Ключові елементи обраного підходу

- Швидка розробка і мінімальний час виходу на ринок.
- Можна вести основну частину розробки в браузері (крім нативної функціональності смартфона - тут знадобиться використовувати смартфон для дебага).
- Можна розробляти додаток для iOS і Android одночасно (з деякими обмеженнями - такими як особливості платформ, що стосуються стилів і плагінів).
- Навички в HTML, CSS і JavaScript - це практично все, що знадобиться для початку розробки. Немає необхідності знати Java і Swift або Objective-C.
- Безліч UI-компонентів доступні і прості у використанні - картки, кнопки, перемикачі, сегменти, попап, поля введення, списки, сітка з рядків і колонок і т. д.
- Безліч плагінів, що дозволяють використовувати функції смартфонів, такі як камера, сканер відбитків пальців, NFC, геолокація, відправка аналітики на Firebase, оповіщення та глибинні посилання.

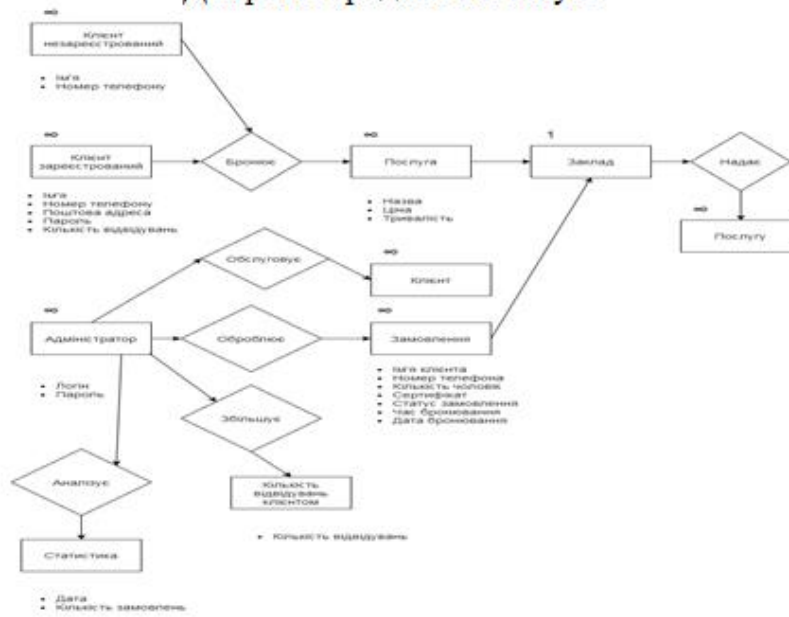
Розгортання системи



Діаграма розгортання

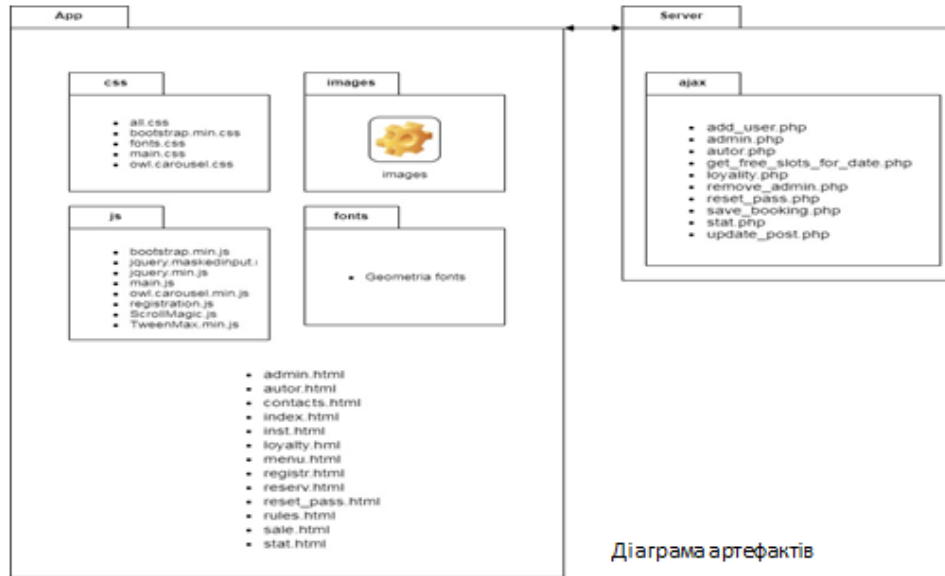
11

Діаграма предметної галузі



12

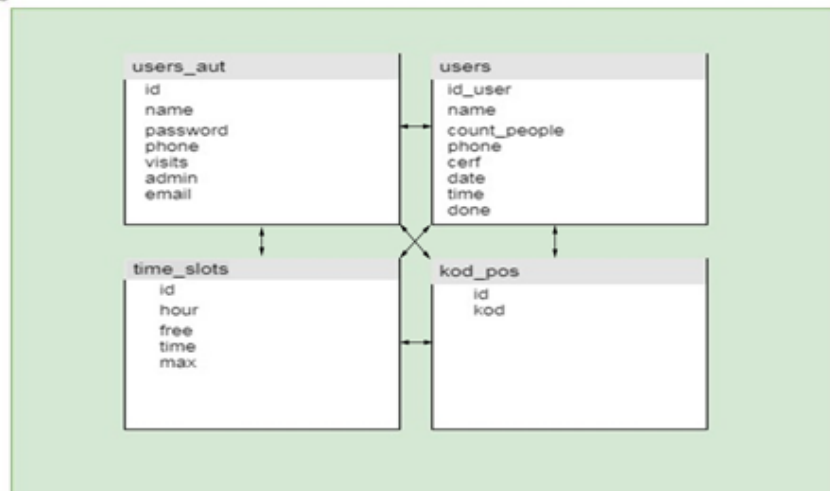
Структура програми



13

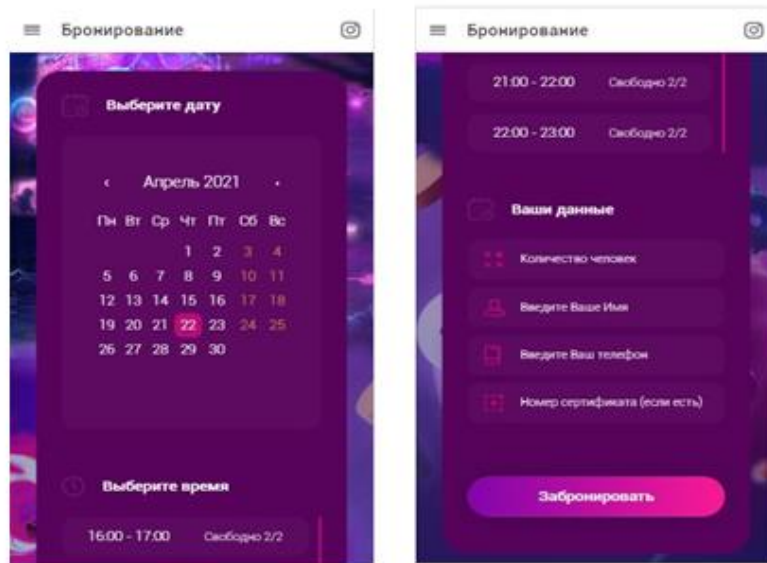
Схема бази даних mysql

ue418458_db



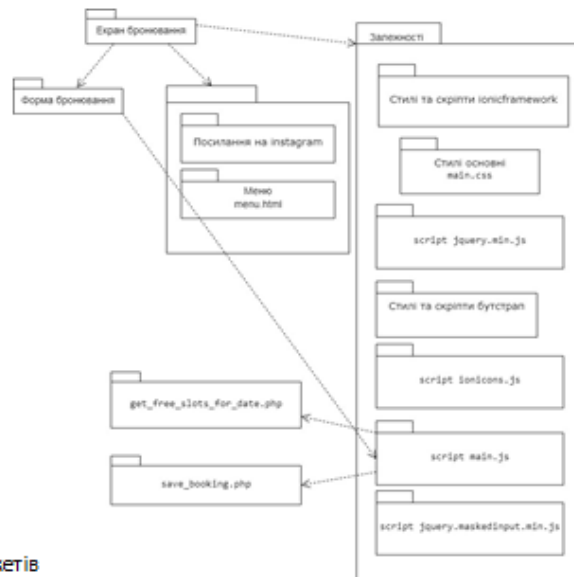
14

Екран бронювання



15

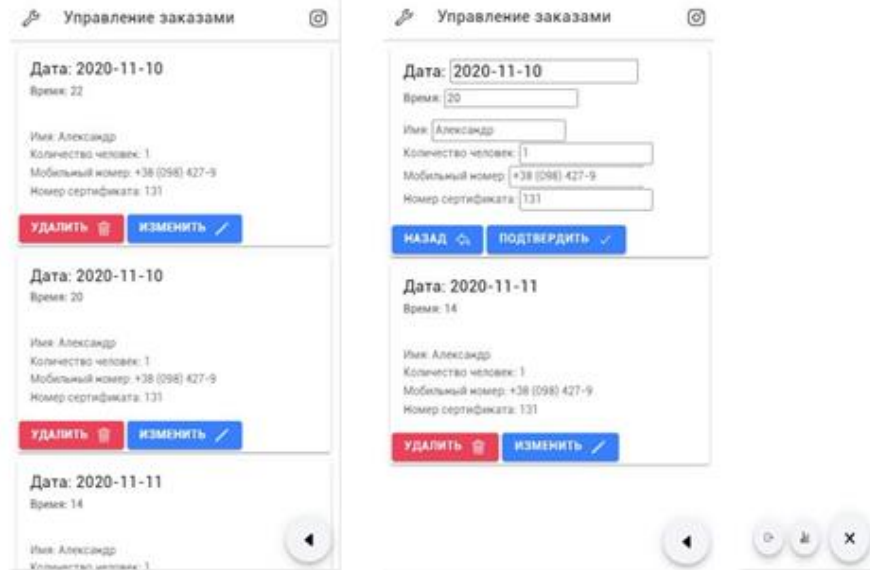
Архітектура підсистеми бронювання



Діаграма пакетів

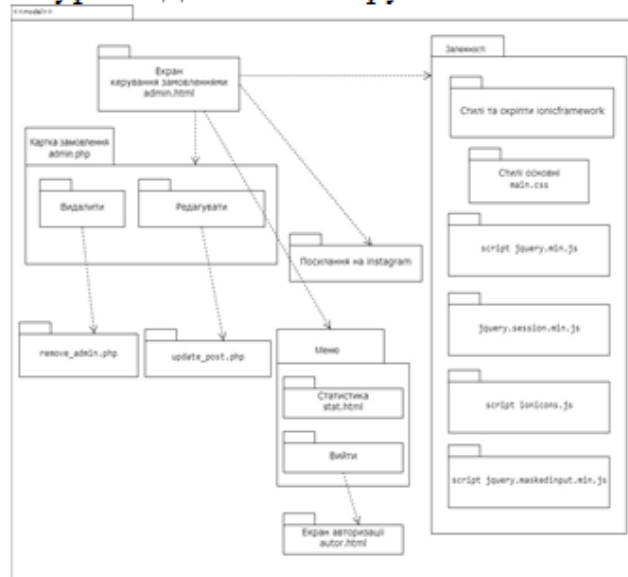
16

Екран керування замовленнями



17

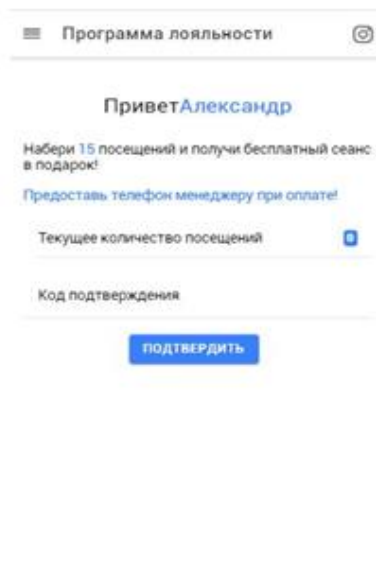
Архітектура підсистеми керування замовленнями



Діаграма пакетів

18

Экран программы лояльности



19

Экран статистики



20

Висновки

Завдання на бакалаврську роботу виконано. Зміст роботи відповідає завданню.

1. Проведено аналіз загальних відомостей про мобільні додатки.
2. Проведено дослідження актуальності розробки мобільного додатку.
3. Проведено аналіз вимог до програмного забезпечення (діаграма прецедентів, діаграма варіантів використання клієнтом, діаграма варіантів використання адміністратором).
4. Виконано моделювання та проектування гібридного мобільного додатку (діаграми пакетів, предметної галузі).
5. Здійснено реалізацію мобільного додатку (діаграми розгортання, артефактів).

Мету роботи досягнуто.

ДЯКУЮ ЗА УВАГУ

