МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені І.І.МЕЧНИКОВА

(повне найменування вищого навчального закладу)

Факультет математики, фізики та інформаційних технологій

(повне найменування інституту, назва факультету (відділення))

Кафедра системного програмного забезпечення та технологій дистанційного навчання

(повна назва кафедри (предметної, циклової комісії))

Дипломна робота

на здобуття освітньо-кваліфікаційного рівня «бакалавр»

(освітньо-кваліфікаційний рівень)

на тему: Розробка мобільного додатку для спільної організації та координації івентів / Development of mobile application for realtime event management

Виконав: студент денної форми навчання
напряму підготовки _123 Комп’ютерна інженерія_
(шифр і назва напряму підготовки, спеціальності)
Швець Вячеслав Вікторович
(прізвище, ім’я, по-батькові)

Керівник ст. викладач Берков Ю. М.
(науковий ступінь, вчене звання, прізвище та ініціали, підпис)

Рецензент к.ф.-м., доцент Шпінарева І.М.
(науковий ступінь, вчене звання, прізвище та ініціали)

Рекомендовано до захисту:	Захищено на засіданні ЕК № ___
Протокол засідання кафедри	протокол № __ від «_ _» ____ 2019 р.
№___ від «___»_____ 2019 р.	Оцінка _____________/______/_____ (за національною шкалою, шкалою ECTS, бали)
Завідувач кафедри	Голова ЕК
____________ (підпис) _Ю.О. Гунченко_ (прізвище, ініціали)	____________ (підпис) _О.О. Арсірій_ (прізвище, ініціали)

Одеса – 2019

АННОТАЦИЯ

В данной дипломной работе разрабатывается мобильное приложение для создания и координации мероприятий на карте в реальном времени.

Цель дипломной работы: создать приложение, позволяющее в реальном времени найти или создать мероприятие на карте, помочь людям с проблемой установления социальных взаимодействий. Особенностью программы является то, что она рассчитана на небольшие краткосрочные мероприятия, которые не получится организовать через механизмы социальной сети Facebook или сервисов, вроде Attendify или Meetup.

В ходе выполнения дипломной работы было разработано мобильное приложение для организации и поиска мероприятий, которое решает поставленную задачу, а также лишено недостатков, которые есть у конкурентов.

АННОТАЦІЯ

У даній дипломній роботі розробляється мобільний додаток для створення і координації заходів на карті в реальному часі.

Мета дипломної роботи: створити додаток, що дозволяє в реальному часі знайти або створити захід на мапі, допомогти людям з проблемою встановлення соціальних взаємодій. Особливістю програми є те, що вона розрахована на невеликі короткострокові заходи, які не вийде організувати через механізми соціальної мережі Facebook або сервісів, на зразок Attendify або Meetup.

В ході виконання дипломної роботи було розроблено мобільний додаток для організації та пошуку заходів, який вирішує поставлене завдання, а також позбавлений недоліків, які є у конкурентів.

ABSTRACT

In this thesis, a mobile application for creating and coordinating activities on a real-time map is being developed.

The aim of the thesis: to create an application that allows you to find or create an event on the map in real time, to help people with the problem of establishing social interactions. A feature of the program is that it is designed for small short-term events that cannot be organized through the mechanisms of the Facebook social network or services like Attendify or Meetup.

In the course of carrying out the thesis, one has developed a mobile application for organizing and searching for events, which solves the problem posed, and also lacks the disadvantages that competitors have.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ 6
ВСТУП 7
1. ОГЛЯД ІСНУЮЧИХ НА РИНКУ РІШЕНЬ 9
1.1 Сервіси для організації заходів 9
Висновки до розділу 11
2. ОГЛЯД ВИКОРИСТОВУВАНИХ ТЕХНОЛОГІЙ................................... 12
2.1 Вибір стеку технологій для розробки додатку 12
2.2 Розробка мобільних додатків на React Native.................................... 13
2.3 Розробка серверних додатків на NodeJS 17
2.4 База даних MongoDB... 20
Висновки до розділу 24
3. РОЗРОБКА КЛІЄНТСЬКОЇ І СЕРВЕРНОЇ ЧАСТИНИ ДОДАТКА 25
3.1 Проектування архітектури сервісу... 25
3.2 Проектування бази даних для додатка 26
3.3 Дизайн клієнтської частини додатку 27
3.4 Розробка аутентифікації через sms 29
3.5 Розробка REST-api 32
3.6 Розробка чату івента 34
3.7 Шер в соціальних мережах 36
Висновки до розділу 39
ВИСНОВКИ.. 40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ... 41
ДОДАТОК А... 42

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ

JSON (JavaScript Object Notation) – це текстовий формат обміну даними між комп'ютерами.

СУБД – система управління базами даних.

БД – База даних.

Фреймворк – інфраструктура програмних рішень, що полегшує розробку складних систем.

Email (email) – адреса електронної пошти.

Аутентифікація – процедура встановлення належності користувачеві інформації в системі пред'явленого ним ідентифікатора.

Шер (share) – процес, у котрому людина ділиться інформацію.

Нативні додатки – прикладні додатки, які були розроблені для використання на певних платформах.

ВСТУП

У сучасному світі комунікаційні технології стали невід'ємною частиною життя людини. Починаючи від месенджерів і закінчуючи соціальними мережами, люди використовують все більше і більше сервісів, спрямованих на соціальну складову їхнього життя.

Соціальні мережі дозволяють користувачам обмінюватися ідеями, цифровими фотографіями і відео, публікаціями та інформувати інших людей про онлайн або реальні події і події з людьми в їх мережі. Успіх соціальних мереж можна побачити в їх домінуванні в суспільстві сьогодні: в 2019 році Facebook має 2,13 мільярда активних користувачів на місяць і в середньому 1,4 мільярда активних користувачів в день. LinkedIn, орієнтована на кар'єру соціальна мережа, як правило, вимагає, щоб учасник особисто знав іншого учасника в реальному житті, перш ніж він зв'яжеться з ним онлайн. Деякі служби вимагають, щоб учасники мали попередньо існуюче з'єднання для зв'язку з іншими учасниками.

Існує безліч різних сайтів і додатків, покликаних зайняти якусь соціально значиму нішу. Наприклад, в Instagram щодня публікуються сотні тисяч фото і відеозаписів, на яких люди показують своє життя, у Twitter'і пишуться сотні тисяч нових постів.

Також існує безліч сервісів, орієнтованих на фахівців в певній галузі. Таких як, наприклад, Dribbble і Behance для дизайнерів, Soundcloud для музикантів, Youtube Для відеомейкерів.

Метою даної дипломної роботи є розробка програми для створення і координації заходів на карті. Програма має складатися з серверної та клієнтської частин. Клієнтска частина призначена для використання на мобильних пристроях.

Принципово ця ідея схожа на те, як працюють івенти в мережі Facebook, проте в додатку, що розробляється, івенти будуть більш гнучкими і

динамічними. Наприклад, навряд чи хтось буде створювати івент в соціальній мережі, якщо він прямо зараз хоче піти побігати в парк, а всі знайомі зайняті. Відсутність додатків, дозволяючих знайти захід поблизу прямо зараз, це проблема. Цей додаток має її вирішити. Користувач, який хоче стати ініціатором заходу в кілька кліків може створити позначку на карті, після чого інші користувачі програми моментально побачать цю позначку на карті у себе, і ті, що зацікавилися зможуть зв'язатися зі створювачем івенту в чаті.

Така модель заходів є більш гнучкою ніж та, що використовується в Facebook та інших аналогах на ринку. Вона передбачена для дрібніших за масштабом заходів, на зразок прогулянок, вечірок, походів в кафе. Швидше за все вона не підійде для організації концерту або виставки в музеї.

В ході роботи над проектом буде розглянуто життєвий цикл продукту, дизайн його інтерфейсу, розробка клієнтської і серверної частини, інтеграція з соціальними мережами.

ВИСНОВКИ

В ході виконання дипломної роботи було розроблено мобільний додаток, який виконує поставлені перед ним задачі, а саме:

- Додаток має зручний і мінімалістичний інтерфейс
- Додаток позбавлено більшості недоліків, які є у конкурентів
- Додаток вирішує проблему встановлення соціальних взаємодій

Також був розроблений дизайн та архітектура серверної частини додатку.

Вдалося уникнути проблем, які є у існуючих на ринку рішень, таких як нав'язливі системи монетизації, які іноді роблять користування ними практично неможлим, диференціювання користувачів за різними ознаками.

Надалі планується розвивати проект і додавати в нього новий функціонал, наприклад:

- Створення приватних заходів
- Тегування заходів
- Анімований інтерфейс
- Додавання користувачів в друзі

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бонни Айзенман, Learning React Native: Building Native Mobile Apps with JavaScript. 2 изд – O'Reilly, 2015.
2. Джакоб Фридман и Эрик Масьелло, Mastering React Native. 1 изд – Packt Publishing, 2017.
3. Expo Docs [Электронный Ресурс] – Режим доступа: https://expo.io/
4. Этан Браун, Изучаем JavaScript. Руководство по созданию современных веб-сайтов. – Вильямс, 2016.
5. Этан Браун, Web Development with Node and Express: Leveraging the JavaScript Stack. – O'Reilly, 2015.
6. Майкл Диролф, MongoDB: The Definitive Guide – O’Reilly. 2013
7. Роберт Мартин, Чистый Код. Создание, анализ и рефакторинг – Prentice Hall. 2008
8. HtmlBook [Электронный ресурс] – Режим доступа: http://htmlbook.ru/.
9. Роберт Мартин, Чистая архитектура. Искусство разработки программного обеспечения – Питер. 2018
10. Илья Бирман, Пользовательский интерфейс Электронный учебник. – Дизайн Бюро Артёма Горбунова, 2016.
11. Account Kit For Developers [Электронный ресурс] – Режим доступа: https://developers.facebook.com/docs/accountkit/
12. React Native Share Docs [Электронный ресурс] – Режим доступа: https://github.com/react-native-community/react-native-share

ДОДАТОК А

ВИХІДНИЙ КОД СЕРВЕРНОГО КОНТРОЛЕРУ ІВЕТНІВ

```
import puppeteer from 'puppeteer'
import EventModel from '../models/Event'
import keepKeys from '../utils/keepKeys'

class EventController {
  create(eventSocketServer) {
    return (req, res) => {
      const fieldsToSave = keepKeys([
        'title',
        'description',
      ], req.body)

      const { location: rowLocation, participants:
rowParticipants } = req.body

      fieldsToSave.creator = req.user.userId
      fieldsToSave.images = [ ...req.files.map(({ filename }) =>
filename) ]
      fieldsToSave.location = JSON.parse(rowLocation)
      fieldsToSave.participants =
[...JSON.parse(rowParticipants), req.user.userId]

      EventModel.create({ ...fieldsToSave })
        .then(event => {
          res.send(event)
          eventSocketServer.emit('EVENT_CREATED', event)
        })
        .catch(err => res.send(err))
    }
  }

  getAll(req, res) {
    EventModel.find({})
      .then(events => res.send(events))
      .catch(err => res.send(err))
  }

  getById(req, res) {
    EventModel.findById(req.params.id)
      .populate('creator')
      .populate('participants')
```

```
      .then(events => res.send(events))
      .catch(err => res.send(err))
  }

  addParticipant(req, res) {
    const { participant } = req.body

    EventModel.findOneAndUpdate(req.params.id, { $addToSet: {
participants: participant } }, { new: true })
      .then(result => res.send(result))
  }

  deleteParticipant(req, res) {
    const { participant } = req.body

    EventModel.findOneAndUpdate(req.params.id, { $pull: {
participants: participant } }, { new: true })
      .then(result => res.send(result))
  }

  renderSocialShareView(req, res) {
    const { query: { eventTitle, eventPicture, eventLocation } }
= req
    res.render(__dirname + '../../views/socialShare.ejs', {
eventTitle, eventPicture, eventLocation })
  }

  async renderSocialShareImage(req, res) {
    const { params: { eventId } } = req
    const { title, images, location } = await
EventModel.findById(eventId)

    const browser = await puppeteer.launch({args: ['--no-
sandbox', '--disable-setuid-sandbox']})
    const page = await browser.newPage()

    await page.goto(`http://localhost:3001/events/${ eventId
}/shareview?eventTitle=${ title }\
      &eventPicture=${ images[0] }\
      &eventLocation=${ JSON.stringify(location) }`)

    await new Promise(res => setTimeout(res, 150)) // wait for
marker to be placed on map

    const screenshot = await page.screenshot({
      type: 'jpeg',
```

```
      quality: 60,
      encoding: 'base64',
      clip: {
        x: 0,
        y: 0,
        width: 1080,
        height: 1920,
      }
    })

    await browser.close()
    res.send(screenshot)
  }
}

export default new EventController()
```