Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)

Кафедра _____Програмної інженерії_____
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

_____другий (магістерський)_____
(рівень вищої освіти)

Дослідження методів інтеграції мобільних пристроїв з веб-орієнтованими інформаційними системами
(тема)

Виконав: студент 2 курсу, групи ІПЗм-17-2___
спеціальності 121- Інженерія програмного забезпечення
(код і повна назва спеціальності)
Освітньо-наукової програми
Інженерія програмного забезпечення_____
(повна назва освітньої програми)

_____Задорожній С. А._____
(прізвище, ініціали)
Керівник ____к.т.н., доцент каф. ПІ Каук В.І._____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____ З.В.Дудар

2019 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121– Інженерія програмного забезпечення
(код і повна назва)

Освітньо-професійна програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Задорожньому Станіславу Андріановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів інтеграції мобільних пристроїв з веб-орієнтованими інформаційними системами
затверджена наказом по університету від "18" квітня 2019 р № 546 Ст
заповнюється вручну після отримання наказу

2. Термін подання студентом роботи до екзаменаційної комісії «___»_____ 2019 р.

3. Вихідні дані до роботи *ОС Windows, середовище об'єктно-орієнтованого проектування Visual Studio, мова програмування C#, алгоритми обробки даних акселерометру, пояснювальна записка, програмна система для інтеграції мобільних пристроїв з ПК.*

4. Перелік питань, що потрібно опрацювати в роботі *мета роботи, аналіз проблемної галузі і постановка задачі, огляд концепції, методів взаємодії мобільного пристрою з веб-браузерами, програмна реалізація програмної системи.*

5. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім’я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	к.т.н., доц. каф. ПІ Каук В.І.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз проблемної області	05.05.2019	Виконано
2	Розробка моделі предметної галузі	10.05.2019	Виконано
3	Розробка структури зберігання даних	15.05.2019	Виконано
4	Створення коду програми	20.05.2019	Виконано
5	Тестування і налагодження програми	01.06.2019	Виконано
6	Підготовка пояснювальної записки	17.06.2019	Виконано
7	Підготовка презентації та доповіді	19.06.2019	Виконано
8	Попередній захист	20.06.2019	Виконано
9	Нормоконтроль, рецензування	21.06.2019	Виконано
10	Занесення диплома в електронний архів	25.06.2019	Виконано
11	Допуск до захисту у зав. кафедри	26.06.2019	Виконано

Дата видачі завдання 22 квітня 2019 р.

Студент _____ Задорожній С. А.
(підпис)

Керівник роботи _____ к.т.н., доц. каф. ПІ Каук В.І.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до атестаційної роботи: 67 с., 16 рис., 20 джерел.

JAVASCRIPT, ВЕБ-ДОДАТКИ, ДАТЧИКИ ОРІЄНТАЦІЇ ПРИСТРОЮ, МОБІЛЬНІ ДАТЧИКИ, МОБІЛЬНІ ПРИСТОРІЇ.

Об’єктом дослідження є методи використання мобільних пристроїв для взаємодії з веб-додатками.

Метою роботи є дослідження можливості використання апаратних пристроїв мобільного телефону для керування веб-додатками або для заміни периферійних пристроїв комп’ютеру можливостями смартфону.

Результати дослідження: проведено аналіз існуючих апаратних пристроїв смартфону, які можуть бути використані для взаємодії з браузером, здійснено аналіз доцільності використання смартфону замість зовнішніх пристроїв для комп’ютеру. Результати роботи можуть бути використані в будь-яких системах, для яких потрібно використовувати зовнішні пристрої.

DEVICE ORIENTATION SENSORS, JAVASCRIPT, MOBILE DEVICES, MOBILE SENSORS, WEB-SYSTEM.

The object of the research is the methods of using mobile devices to interact with web applications.

The aim is to explore the possibility of using mobile phone hardware devices for control web applications or for replacing peripheral devices with computer smartphone sensors.

The result of the research: existing hardware devices on the smartphone, which can be used to interact with the browser has been analyzed and the feasibility of using a smartphone instead of external devices for the computer has been analyzed. The result of the work can be used on any system for which external devices are required.

ЗМІСТ

ВСТУП

Сьогодні дуже важко уявити собі повсякденне життя людини без використання смартфону. Практично всі звичні нам функції з'явилися в телефонах ще на рубежі століть. У 1999 році пристрої навчилися виходити в інтернет по протоколу WAP, згодом швидкість зросла з 10 кілобайт в секунду до декількох мегабайт. Всі інші характеристики телефонів еволюціонували. Візьмемо як приклад вбудовані камери - перша з них мала 0,3 мегапікселя, а зараз на ринку можна знайти апарати навіть з 41 Мпікс.

Телефони перетворюються в повноцінні портативні комп'ютери, до яких можна підключати зовнішні монітори, клавіатури і миші. У них великий обсяг оперативної пам'яті (вже зараз є восьміядерні процесори з RAM більше 8 Гб).

Сучасний смартфон - це повноцінна вимірювальна станція, про яку в шістдесятих вчені не могли і мріяти [1].

Смартфони доповнюють усі сфери нашого життя. За допомогою телефону зараз можна розрахуватися за покупку у магазині, підтвердити свою особистість при виконання банківських операцій або при авторизації у будь-якому мобільному додатку, зробити якісні фотографії на рівні професіональних фотокамер, виміряти освітленість приміщення, використовувати телефон у якості компасу та багато іншого. Усі ці функції присутні майже у кожному звичайному бюджетному телефоні і вони завжди під рукою і готові для використання.

Але якщо людина хоче провести час за звичайним комп’ютером, то всі доступні функції смартфонів втрачають свою актуальність. Для спілкування у відео чатах необхідно купити додаткову відеокамеру спеціально для персонального комп’ютера, хоча у смартфоні наявна набагато краща камера. Знаходячись перед екраном монітору у користувача немає можливості використовувати свої відбитки пальців, Face ID, сітківку ока для авторизації на веб-сайті. Усі наявні можливості смартфону неможливо зараз використовувати для взаємодії зі своїм комп’ютером, тому виникає необхідність купувати додаткові пристрої, які повністю дублюють

функціонал смартфону, який завжди під рукою. Також усі додаткові пристрої необхідно під’єднувати до комп’ютера за допомогою USB кабелів, що зменшує кількість доступних портів, які можливо використовувати для підключення інших необхідних пристроїв.

Саме тому актуальною виникає тема використання мобільних пристроїв з багатою внутрішньою складовою для роботи користувачів з персональними комп’ютерами.

Метою даної роботи є дослідження можливості використання мобільних пристроїв для взаємодії з веб-додатками та дослідження можливих шляхів та доцільності використання смартфонів замість незалежних зовнішніх пристроїв, які можливо підключати до комп’ютера.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної галузі

Мобільна революція повністю змінила спосіб доступу людей до Інтернету. Звичайно, що веб має розвиватися, щоб краще підходити до численних пристроїв, на яких він доступний і зараз.

Коли веб-сторінка була створена 25 років тому, вона була розроблена спеціально для персональних комп’ютерів. За ці 25 років ми бачили, як веб-сайт поширюється на настільні ПК, ноутбуки, мобільні телефони, планшети, а тепер навіть на модні аксесуари, як годинник.

Тому за цей час почали розроблюватися безліч бібліотек для взаємодії мобільного браузеру з різноманітними сервісами смартфонів. Але доступ до всіх сервісів через браузер неможливий через міркування безпеки, адже якщо зловмисник почне використовувати на своєму веб-додатку з недобрим наміром датчики телефону, то це може дуже сильно нашкодити пристрою або даним користувача. Отже для того, щоб зрозуміти які зовнішні пристрої можливо замінити на смартфон необхідно спочатку розглянути, які все ж таки сервіси доступні браузеру мобільного телефону за допомогою JavaScript.

Ще в липні 2009 року в рамках World Wide Web Consortium (W3C) була створена "Device APIs Working Group (GAP)" [2], метою якої є створення client-side API для взаємодії веб-додатків з сервісами пристроїв (камера, календар, контакти, орієнтація пристрою та ін.). Місія робочої групи «Devices and Sensors Working Group» (DAS WG) полягає у створенні API на стороні клієнта, яке дозволить розвивати веб-додатки, які взаємодіють з апаратними засобами пристроїв і датчиками.

Робоча група прагне зібрати консенсус щодо його специфікацій, як у межах цієї групи, так і спільноти в цілому, працюючи через ітеративний процес з виконавцями, щоб забезпечити реалізацію та впровадження специфікацій.

Робоча група була вперше затверджена у липні 2009 року як робоча група «Device APIs & Policy Working Group», як продовження семінару з питань безпеки для доступу до API пристроїв з Інтернету, що відбувся у грудні 2008 року, потім перейменовані у серпні 2011 року під назвою «Device APIs Working Group», і знову перейменовані в березні 2016 року під назвою «Device and Sensors Working Group».

На даний час вже розроблено і доступно для використання декілька сенсорів [3], а саме:

– Battery Status API. API стану батареї надає інформацію про рівень або стан батареї системи. Завдяки цьому API можна дізнатися, чи батарея заряджається чи ні, як скоро батарея повністю розрядиться, або просто її поточний рівень. Ці дані доступні через чотири властивості, які належать об'єкту window.navigator.battery. Цей API також визначає події, які можуть бути запущені при зміні стану батареї. Даний API може бути використаний наприклад у даному випадку: користувач працює над документом, використовуючи веб-додаток, на телефоні який майже розряджений і телефон раптово вимикається без можливості зберегти внесені зміни. Завдяки цьому API веб-додаток може бути розроблений таким чином що він буде здатний виявляти поточний рівень батареї і частіше зберігати зміни, щоб запобігти втраті даних у разі низького або критичного значення заряду акумулятора [4].

– Vibration API. Призначений для управління вібросигналом пристрою.

– Screen Orientation API. Призначений для отримання подій зміни орієнтації екрану пристрою, інформації про поточний стан орієнтації екрану.

– Device Orientation API. Призначений для отримання подій зміни орієнтації пристрою. Виявлення орієнтації пристрою корисно для широкого кола випадків, від навігації до ігор. Цей API визначає кілька подій, які надають інформацію про фізичну орієнтацію та рух пристрою. API надає наступні три події deviceorientation, devicemotion і compassneedscalibration. Перша запускається, коли акселерометр виявляє зміну орієнтації пристрою. Друга спрацьовує щоразу, коли пристрій прискорює або сповільнюється. Остання подія виникає, коли агент користувача визначає, що компас вимагає калібрування [5].

– Device Motion API. Призначений для отримання подій датчика-акселерометра про переміщення пристрою.

– Ambient Light Events. Призначені для отримання подій датчика освітленості пристрою.

– Proximity Events. Події датчика наближення пристрої.

– Page Visibility API. Дозволяє визначити чи відображається сторінка на екрані пристрою.

– Fullscreen API. Надає можливості роботи з повноекранним режимом пристрою.

– Geolocation API. API геолокації дозволяє визначити місцезнаходження користувача за допомогою можливостей позиціонування свого пристрою. Велику частину часу це буде зроблено за допомогою GPS, але також можуть бути використані менш точні методи, такі як позиціонування на основі WiFi.

– Media Capture and Streams API. За допомогою API Media Capture і Streams можна отримати доступ до камери пристрою за допомогою JavaScript. У поєднанні з потужністю WebRTC це дає змогу застосувати деякі цікаві програми, такі як відеоконференції в реальному часі без необхідності використання сторонніх плагінів.

Але окрім цього списку існує й можливість використовувати інші сенсорні датчики пристрою за допомогою нативних додатків для конкретної операційної системи. Проте такий підхід змушую розробників писати декілька додатків під кожну ОС окремо, що збільшує витрати на розробку там підтримку даного програмного забезпечення.

Спираючись, на доступний набір можливостей можна виділити декілька датчиків, які будуть використання для подальшого розгляду у даній роботі.

Такими API були обрані Device Motion and Orientation API та Media Capture and Streams API.

1.2 Постановка задачі

Відповідно до аналізу предметної галузі було вирішено розробити декілька додатків, які б задовольняли наступним вимогам і у повному обсязі могли би продемонструвати роботу обраних сенсорів:

– Веб-додаток для підключення камери смартфону до комп’ютера та транслювання сигналу з камери на комп’ютер. Після підключення камери до цього додатку користувач матиме змогу використовувати цю камеру на комп’ютері на кшталт реальної веб-камери приєднаної до робочої станції по USB.

– Веб-додаток для підключення датчиків акселерометру смартфону до комп’ютера та дистанційне керування курсором миші на комп’ютері або прокручування сторінки на комп’ютері за допомогою жестів на телефоні.

У якості функціоналу додаток має підтримувати наступні функції:

– надавати можливість встановлення з’єднання з комп’ютером за допомогою QR коду;

– керувати мишкою комп’ютера за допомогою датчиків мобільного пристрою;

– транслювати відео з мобільного пристрою у якості віртуальної камери комп’ютера.

Також дана програмна система повинна бути розроблена для комп’ютерів з ОС Windows 10 та для смартфонів на базі ОС Android.

2 ОПИС ДОСЛІДЖЕНЬ ТА ПОРІВНЯННЯ ІСНУЮЧИХ АНАЛОГІВ

За результатами аналізу предметної галузі було обрано датчик орієнтації пристрою та камера для розробки програмної системи для взаємодії смартфону та персонального комп’ютеру.

За допомогою датчику орієнтації телефону можна керувати мишкою комп’ютера на відстані, такі функції може виконувати такий фізичний пристрій як пульт. Даний пристрій може бути використаний наприклад для перемикання слайдів на комп’ютері під час презентації. А за допомогою камери смартфону можливо замінити веб-камеру на комп’ютері.

Для порівняння даних пристроїв з розроблюваним додатком та визначення доцільності заміни пристроїв на датчики смартфону було обрано наступні критерії:

– швидкість передачі даних;
– ціна;
– додаткові вимоги для використання даної функціональності;
– безпечність.

Використовуючи дані критерії обидва пристрої можуть бути порівняні одночасно, бо вона мають схожі ознаки за даними параметрами.

Швидкість передачі даних за допомогою спеціальних пристроїв буде більшою ніж при передачі даних через інтернет, але у наш час швидкість інтернету постійно збільшується, тому не має бути великих затримок при передачі сигналу.

Зараз майже кожна людина має смартфон, тому усі доступні сенсори вже вбудовані в нього і не потребують додаткових витрат для їх використання, а зовнішні пристрої навпаки потребують додаткових коштів для їх придбання.

Також використання датчиків смартфонів не потребують ніяких додаткових налаштувань, окрім встановлення спеціальної програми для взаємодії з смартфоном, а зовнішні пристрої потребують використання додаткового порту USB, що зменшує кількість доступних портів. У даний час це може викликати певні незручності, адже кількість таких портів обмежена.

Передача сигналу по дроту для зовнішніх пристроїв є набагато безпечною ніж по інтернету. Але при використанні зашифрованих каналів (ssl) у зловмисників практично немає шансів розшифрувати перехвачений сигнал. Тому обидва підходи є повністю безпечними.

Отже, порівнявши пристрої по заданим критеріям можна прийти до висновку, що використанням сенсорів телефону поступається лише можливою швидкістю передачі даних, але при нормальному підключенні до мережі дана проблема не повинна впливати на результат. Тому доцільно розробити програмну систему для заміни фізичних пристроїв на використання сенсорів телефону.

2.1 Історія розвитку сенсорів в мобільних пристроях

Технологічні досягнення та величезна інтеграція сенсорів у мобільних телефонах збільшили важливість мобільних телефонів у житті людей та перетворили їх роль з простого пристрою зв'язку в ящик Пандори. Ряд сервісів було винайдено різноманітними співтовариством, організаціями та науковими колами, які використовують сенсорні можливості мобільних телефонів і отримують величезну кількість інформації про навколишнє середовище або про людей, які можуть бути використані продуктивно у вирішенні реальних проблем.

Сенсори розвиваються як життєво важливі джерела даних в Інтернеті, переконуючи організації досліджувати нові методи побудови бездротових сенсорних мереж, а також захоплювати, обробляти, аналізувати та використовувати дані, що генеруються датчиками. Розуміння переваг сенсорів мобільних телефонів над бездротовими датчиками та мініатюризація в технологіях датчиків проклали шляхи інтеграції сенсорів у мобільні телефони безпосередньо або через технології бездротових мереж (наприклад, Bluetooth тощо). Використання датчиків у мобільних телефонах перетворить мобільні телефони в зонди. Використання мобільних телефонів як датчиків має ряд практичних переваг

у порівнянні з традиційними бездротовими сенсорними мережами. Мобільні телефони завжди супроводжуються користувачами, тому вирішує проблеми енергоменеджменту, формування та обслуговування мережі. Велику частину навантаження на обслуговування можна усунути від оператора або адміністратора мережі, оскільки кінцевий користувач завжди проявляє велику зацікавленість у підтримці своїх мобільних телефонів, таких як ремонт обладнання, встановлення або видалення програмного забезпечення та підтримка управління даними з серверами, тощо. Використання мобільних телефонів як сенсорів замість звичайних сенсорів, очевидно, мало б високу економію.

Сенсори в мобільному телефоні можна розділити на фізичні сенсори та віртуальні датчики. Фізичні датчики є апаратними датчиками, вбудованими безпосередньо в пристрої мобільних телефонів, і отримують їх дані безпосередньо шляхом вимірювання конкретних навколишніх характеристик (наприклад, акселерометра, гіроскопа, датчика близькості, тощо). Віртуальні датчики є програмними датчиками, які отримують свої дані з декількох апаратних сенсорів (наприклад, в платформі Android лінійне прискорення і датчики гравітації є віртуальними датчиками). Більшість сучасних смартфонів є відкритими та програмованими і мають ряд вбудованих датчиків. Кількість та типи датчиків у мобільних телефонах змінюються залежно від базових платформ мобільних телефонів і зручності використання. Розуміння потенціалу датчиків і збільшення мініатюризації в технологіях дозволить інтегрувати в майбутні мобільні телефони у більш просунуті датчики. Деякі з найпопулярніших вбудованих датчиків мобільних телефонів:

Датчик відстані. Датчик відстані випускає електромагнітне або електростатичне поле або промінь електромагнітного випромінювання (наприклад, інфрачервоний) і шукає зміни у польовому або зворотному сигналі та виявляє будь-які поблизу присутні об'єкти без будь-якого фізичного контакту. У смартфонах датчик відстані відчуває близькість екрана телефону до тіла користувача, коли телефон розташований поблизу вуха. Датчик відстані автоматично вимикається та блокує екран не тільки для економії заряду акумулятора, але й для захисту від

небажаного дотику до екрану. Як тільки мобільний телефон віддаляється від вуха, він автоматично відновиться до попереднього стану.

Датчик навколишнього освітлення. Цей датчик вимірює світло навколишнього середовища і відповідно регулює яскравість мобільного телефону, щоб оптимізувати видимість екрана. Якщо світло навколишнього середовища високе, яскравість екрану мобільного телефону зменшиться в іншому випадку. Регулювання яскравості дисплея не тільки оптимізує видимість, але й заощадить заряд акумулятора в смартфонах.

Датчик акселерометра. Датчик акселерометра використовується як контролер інтерфейсу користувача: зміна дисплея екрана шляхом визначення орієнтації пристрою на основі способу розташування пристрою користувачем. Цей датчик вимірює прискорення смартфона в трьох різних осях: X, Y і Z. Смартфони зазвичай використовують 3-осьовий акселерометр для виявлення орієнтації телефону і відповідно регулюють відображення екрану, забезпечуючи зручність для користувачів, які легко перемикаються між портретом і пейзажем. Крім того, камера також використовує акселерометр, щоб визначити, чи слід робити знімки в портретному або ландшафтному режимах [6]. Apple iPhone 4 також використовує датчик гіроскопа в поєднанні з акселерометром для ефективного сприйняття способу переміщення пристрою. Акселерометри можуть широко використовуватися в багатьох системах зондування мобільних телефонів, включаючи визнання діяльності користувачів, моніторинг дорожнього руху та доріг, автоматичне виявлення ДТП тощо.

Датчик цифрового компаса. Цей датчик є магнітометром, який використовується для розпізнавання півночі і визначає напрямок для користувачів. Сучасні смартфони не використовують магніти, оскільки магнітні перешкоди можуть значно погіршити можливості стільникового зв'язку через зниження рівня сигналу. Наприклад, iPhone 4 від Apple використовує чіп AKM AK8975, який є невеликим датчиком і визначає орієнтацію і напрямки відносно атмосфери Землі за допомогою ефекту Холла.

Усі зазначені вище сенсори можна віднести до мікроелектромеханічних систем.

Мікроелектромеханічні системи (Microelectromechanical systems, МЕМС) представляють собою мініатюрні пристрої, що містять мікроелектронні і мікромеханічні компоненти. Кожен день людство користується безліччю девайсів, заснованих на базі цих рішень. Найпростішим прикладом мікроелектромеханічних системи може служити акселерометр, який використовується у всіх сучасних смартфонах, ігрових консолях і жорстких дисках. Однак існує безліч інших систем, застосування яких аж ніяк не обмежується споживчою електронікою. Рішення на основі МЕМС знаходять застосування в автомобільній промисловості, військової галузі, а також медицині.

Початком розвитку МЕМС можна вважати 1954 рік. Саме тоді було відкрито п'єзорезистивного ефект кремнію і германію, який ліг в основу перших датчиків тиску і прискорення. Через 20 років - в 1974 році - компанією National Semiconductor вперше було налагоджено масове виробництво датчиків тиску. А в 1990-х роках ринок мікроелектромеханічних систем значно виріс завдяки початку використання різних мініатюрних сенсорів в автомобільній електроніці.

MEMS-системи отримали приставку «мікро» через своїх розмірів. Складові частини таких пристроїв мають розміри від 1 до 100 мкм, а розміри готових систем варіюються від 20 мкм до 1 мм.

У плані архітектури МЕМС-пристрій складається з декількох взаємодіючих механічних компонентів і мікропроцесора, який обробляє дані, одержувані від цих компонентів. Якогось стандарту для механічних елементів немає: за своїм типом вони можуть сильно відрізнятися в залежності від призначення конкретного пристрою.

Як матеріали для виробництва МЕМС можуть використовуватися як і традиційний кремній, так і інші матеріали: наприклад, полімери, метали і кераміка. Найчастіше механічні системи виготовляються з кремнію. Його основні переваги полягають в фізичних властивостях. Так, кремній дуже надійний - він може працювати протягом трильйонів циклів операцій і при цьому не руйнуватися. Що

стосується полімерів, то цей матеріал хороший тим, що його можна виробляти у великих кількостях і, що найважливіше, з безліччю різних характеристик під конкретні завдання. Ну а метали (золото, мідь, алюміній), в свою чергу, забезпечують високі показники надійності, хоч і поступаються за якістю своїх фізичних властивостей кремнію.

Варто окремо згадати і про такі матеріали, як нітрид кремнію, алюмінію і титану. Завдяки своїм властивостям вони широко використовуються в мікроелектромеханічних системах з п'єзоелектричної архітектурою.

Що стосується технологій виробництва МЕМС, то тут використовується кілька основних підходів. Це об'ємна мікрообробка, поверхнева мікрообробка, технологія LIGA (Litographie, Galvanoformung і Abformung - літографія, гальваностегія, формування) і глибоке реактивне іонне травлення. Об'ємна обробка вважається самим бюджетним способом виробництва МЕМС. Її суть полягає в тому, що з кремнієвої пластини шляхом хімічного травлення видаляються непотрібні ділянки матеріалу, в результаті чого на пластині залишаються тільки необхідні механізми.

В цілому, всі МЕМС можна розділити на дві великі категорії: сенсори і актуатори. Розрізняються вони принципом своєї роботи. Якщо завдання сенсора полягає в перетворенні фізичних впливів в електричні сигнали, то актуатор виконує прямо протилежну роботу, переводячи сигнал в будь-які дії. Той же акселерометр є сенсором, а як приклад пристрою, що використовує актуатори, можна привести DLP-проектор

Найпоширенішим МЕМС-пристроєм є акселерометр. Як вже говорилося вище, сфера його використання надзвичайно широка. Вона охоплює мобільні телефони, ноутбуки, ігрові приставки, а також більш серйозні пристрої, такі як автомобілі. Саме призначення акселерометра полягає в вимірі удаваного прискорення. У випадку з мобільними телефонами він використовується для багатьох цілей. Наприклад, для зміни орієнтації екрану. Або ж виконання будь-яких функцій при «струшуванні» пристрою. Крім цього, не варто забувати і про ігри - вони, мабуть, складають основну сферу застосування акселерометрів. Нині

вже складно уявити «просунуту» іграшку, в якій не було б реалізовано управління за допомогою нахилу телефону. Одним словом, акселерометр став невід'ємною частиною смартфонів. До речі, вперше він був встановлений в мобільний телефон Nokia 5500. Завдяки акселерометру телефон можна було використовувати як крокомір [7].

Але незважаючи на те, що рамки використання акселерометра досить чітко визначені, розробники продовжують думати над тим, в яких ще цілях можна застосовувати цей пристрій. Наприклад, вчені з Національного інституту геофізики і вулканології Італії Антоніо Д'Аллесандро і Джузеппе Д'Анна запропонували використовувати акселерометр мобільного телефону як датчик землетрусів. Дуже цікаво! Дослідження проводилися з акселерометром iPhone, і результати порівнювалися з показаннями повноцінного датчика землетрусів компанії Kinemetrics. Як виявилося, мобільний гаджет здатний вловлювати сильні землетруси магнітудою понад 5 балів за шкалою Ріхтера, але тільки якщо він знаходиться поблизу епіцентру підземних поштовхів. Результати не так вражають, однак вчені впевнені: чутливість акселерометрів буде тільки зростати, і в майбутньому вони зможуть визначати і менш сильні землетруси. Можливість акселерометру телефону вимірювати силу підземних поштовхів замість датчиків землетрусу може надати великі переваги у майбутньому. Вся справа в тому, що вчені ставлять собі за мету створення в майбутньому цілої мережі з смартфонів в сейсмічно активних районах. У теорії, при землетрусах дані зі смартфонів будуть надходити в аналітичний центр, що дозволить визначати найбільш постраждалі від стихії райони і правильно координувати рятувальні операції. Ідея більш ніж цікава і, головне, дійсно затребувана в деяких куточках світу, проте зараз складно уявити, як вона буде реалізована на практиці.

Особливе значення має акселерометр, який застосовується в ноутбуках, а точніше, в їх жорстких дисках. Всім відомо, що вінчестери - пристрої досить крихкі, і в разі з лептопами ймовірність їх пошкодження зростає в рази. Так, при падінні ноутбука акселерометр фіксує різке зміна прискорення і віддає команду на

парковку головки жорсткого диска, запобігаючи і пошкодження пристрою, і втрату даних.

В цілому найбільшим і перспективним ринком для акселерометрів та інших МЕМС є автомобільна промисловість. Справа в тому, що на відміну від ринку мобільних та ігрових пристроїв, де акселерометри використовуються в розважальних цілях, в автомобілях на роботі акселерометра грунтуються буквально все системи безпеки. З їх допомогою працюють система розгортання подушок безпеки, антиблокувальна система гальм, система стабілізації, адаптивний круїз-контроль, адаптивна підвіска, система Traction Control. З огляду на, що виробники автомобілів приділяють особливу увагу безпеці, кількість застосовуваних акселерометрів та інших МЕМС буде лише зростати.

2.2 Використання датчиків смартфонів в існуючих веб-сайтах

Світ веб-дизайну яскравий і різноманітний. Великі зміни відбуваються тут не щороку. Підтверджує це і стрімке зростання технологій, який за останні кілька років показав, що важливі тенденції в веб-дизайні недалекого майбутнього будуть зосереджені на поліпшенні наявних розробок.

Плоский дизайн стане більш текстурированним, "кінематографічні досліди" будуть більш поширеними, а використання бібліотек JavaScript дозволить більше експериментувати. Програмісти впевнені, що зростаюча популярність WebGL, віртуальної реальності та використання датчиків пристроїв дозволить змінити вебсайти, які користувачі знають, в щось нове з цікавими інтерактивними можливостями.

Підтримка використання показників акселерометра і гіроскопа за допомогою javascript - це технологія, яка випередила час. Тоді, в далекому 2010р., мобільний інтернет не був так розвинений. Адаптивність верстки була обов'язковим пунктом, та й взагалі сайти були призначені в основному для перегляду на звичайних

моніторах. Зараз же все по-іншому, і частка мобільного трафіку складає мало не 50%, але дана технологія використовується не дуже часто.

Сучасні мобільні пристрої імют не тільки потужні обчислювальні характеристики, але і володіють різноманітністю різного роду датчиків, що дозволяють дізнатися як деякі властивості поточного середовища, так і деякі властивості самого пристрою в цьому середовищі. Для визначення характеристик пристрою в просторі в більшості випадків використовуються два датчика, встановлених на більшості сучасних мобільних пристроїв - це акселерометр і гіроскоп. Показання цих датчиків можуть зчитуватися не тільки нативними додатками, але і web-сторінками, за допомогою javascript. Причому, не потрібно ніяких дозволів, запитів, винятків безпеки та іншого - свідчення можна зчитувати відразу ж після завантаження веб-сторінки. Програміст може отримувати дані про прискорення пристрою по трьох осях, про обертання пристрою за тими ж осями і про прискорення з урахуванням гравітації. Так, на сьогоднішній день можна зустріти кілька цікавих застосувань даних можливостей при веб розробці веб-сайтів.

Створення паралакс ефекту при нахилах смартфона вправо-вліво. Так само використовується поєднання показників датчиків і css-фільтрів. Коли телефон лежить горизонтально на столі - відображається чітке зображення, але як тільки користувач починає нахиляти його на себе - зображення починає розмиватися і при досягненні апаратом 45 градусів плавно з'являється напис. Також можна використовувати і інші фільтри - вибір обмежений тільки підтримкою фільтрів браузерами.

Також при поєднанні фільтрів і гіроскопа можна використовувати відтінки. Коли телефон лежить на столі, обертання його в площині столу буде змінювати колір картинки, яка відображена на дисплеї, як якщо б користувач переміщував повзунок hue/saturation в фотошопі. Таким же чином можна додати зміну яскравості і насиченості при обертанні за двома іншими осями.

Ще одним прикладом використання акслерометра в веб-розробці є слайдер, керований нахилом телефону. При нахилах пристрою вліво-вправо - картинки

рухатимуться в ту чи іншу сторону. Його можна прив'язати до будь-якого існуючого слайдеру або каруселі, і користуватися їх ефектами.

Узагальнюючи розвиток веб-стандартів в сфері фронтенд-технологій (css, js, html), можна побачити загальну тенденцію інноваційних розробок, кінцевою метою яких є становлення веб-інтерфейсів як стандарту де-факто в якості інтерфейсів додатків.

2.3 Порівняння існуючих аналогів

Ідея використання мобільного пристрою для керування комп’ютером виникала вже дуже давно, і з швидким розвитком смартфонів кількість датчиків в середині тільки збільшується тому і сфера використання мобільного пристрою при взаємодії з ПК тільки збільшується. Тому зараз існують деякі аналоги, які частково реалізують даний функціонал [8]. Але серед усіх досліджуваних додатків не було знайдено можливості керувати курсором миші за допомогою датчиків орієнтації пристрою, а лише за допомогою жестів.

Найпопулярніший додаток у цій категорії є Unified Remote (див. рис. 2.1). Ця програма, як і більшість з розглянутих далі, представлена в Google Play в безкоштовній та платній версіях. Але навіть у своїй базовій функціональності вона може з легкістю замінити мишу і клавіатуру і пропонує користувачеві кілька розширених можливостей по управлінню популярними програмами та ПК. Сам додаток не виділяється вишуканим дизайном, хоча відповідає концепції Material Design. Основна увага була приділена режимам управління, так званим «пультів» (Remotes), які зручно згруповані по найменуванню програми для ПК або завданням використання. У безкоштовної версії утиліти користувачеві доступно 18 пультів. Крім режиму для управління мишею і клавіатурою, програма містить власну повноекранну клавіатуру, окремі блоки мультимедійних, цифрових і функціональних клавіш. Платна версія програми містить додатково понад 40

шаблонів управління (включаючи створені користувачем власноруч), підтримку голосових команд, можливість створювати віджети і швидкі дії. Крім того, власники смартфонів з ІЧ-портом зможуть перетворити свій девайс в справжній універсальний пульт завдяки підтримці ІЧ і відповідного режиму управління. В цілому, Unified Remote залишає дуже хороше враження, а його функціональні можливості дійсно вражають.

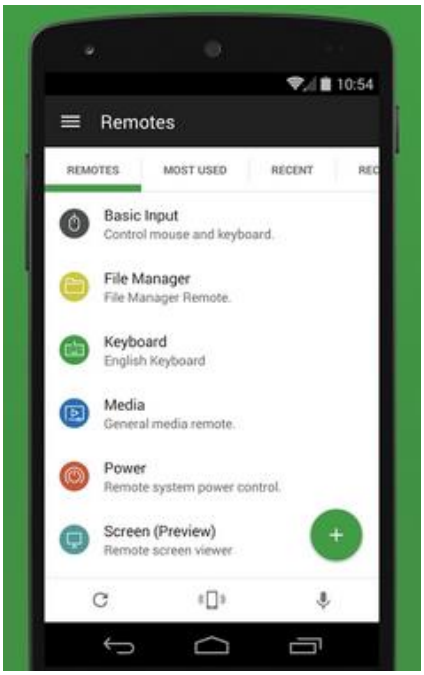

Рисунок 2.1 – Інтерфейс користувача додатку Unified Remote

Утиліта Max Remote (див. рис. 2.2) пропонує чималу кількість готових схем віддаленого управління. Додаток виглядає дуже аскетично, тим не менш, інтерфейс цілком зручний, а всі елементи легко доступні користувачеві. Можливості Max Remote включають в себе управління мишею, слайдами, браузером, живленням. У додатку є готові шаблони для запуску на ПК деяких популярних програм і передбачено командний рядок. Кожна з функцій супроводжується короткою інформацією про її призначення. Єдине, чого немає в Max Remote - це повноцінної підтримки клавіатури, хоча цифровий блок і передача тексту, як додаткова опція, тут присутня. Особливістю програми є емуляція різних видів геймпадів, включаючи найбільш популярні для ігрових консолей. У налаштуваннях програми

користувач може асоціюватися кнопки з діями контролера і змінити значення прискорення, точності і частоти поновлення курсора миші.

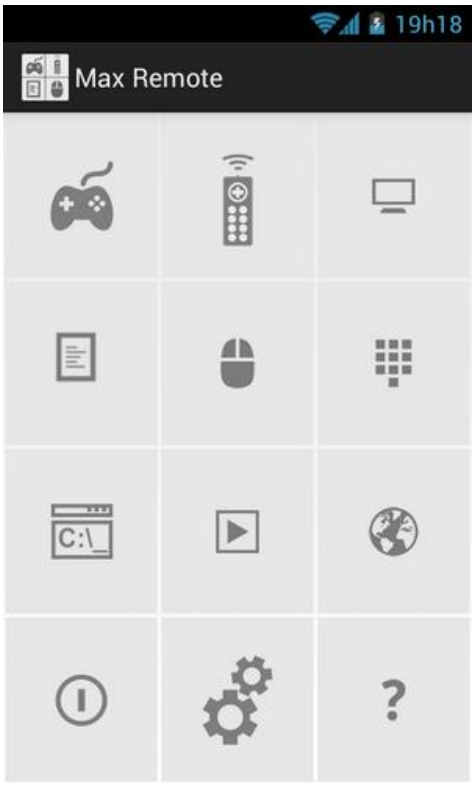

Рисунок 2.2 – Інтерфейс користувача додатку Max Remote

Програма PC Remote (див. рис. 2.3) з'явилася в Google Play відносно нещодавно. Спочатку додаток розроблявся для платформи Windows Phone і саме там став популярним. На Android утиліта знаходиться в статусі бета-версії і поки що не може похвалитися повноцінною функціональністю. На поточний момент в програмі відсутні кілька цікавих опцій, таких як можливість переглядати і передавати файли з ПК і на ПК, створювати віртуальні бездротові мережі на ПК і підключати до нього мобільний пристрій. PC Remote дозволяє підключитися до комп'ютера через Wi-Fi з'єднання за умови знаходження клієнта і сервера в одній мережі. У найближчих оновленнях розробники планують додати можливість підключення через Інтернет. Схеми управління в додатку розбиті за трьома категоріями, які відповідають за базовий введення, системну гучності, функції живлення комп'ютера, керування презентацією Power Point і аудіо-, відеоплеєра. Що стосується емуляції основних пристроїв введення інформації, то реалізація їх в

програмі виявилася не зовсім вдалою, часом незручною. Режим миші являє собою область тачпада і скролінгу. Налаштувань по чутливості і швидкості руху курсора в додатку немає. Клавіатура як така тут відсутня, є тільки блок функціональних клавіш і деякі часто використовувані комбінації, такі як «скопіювати», «вирізати», «вставити» та інші. Процес набору тексту відбувається за допомогою системної клавіатури Android.

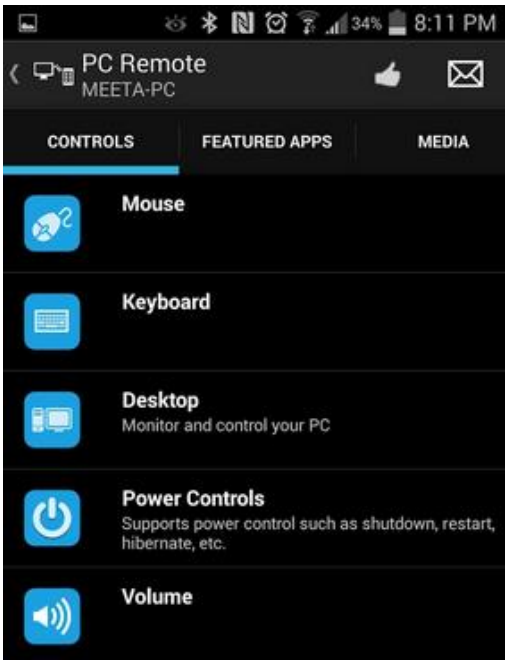

Рисунок 2.3 – Інтерфейс користувача додатку PC Remote

Серед мінусів усіх оглянутих додатків можна зазначити те, що вони потребують встановлення додатку на смартфон, тому неможливо користуватися даними програми на пристроях з іншою ОС. Також одним з важливих недоліків є те, що усі додатки потребують того, щоб комп'ютер та мобільний телефон знаходилися у одній мережі. І також серед зазначених аналогів немає жодного додатку, який би поєднував усі зазначенні функції(камера та керування мишкою) у одному додатку.

Отже, запропонована тема є дуже актуальною і фактично не має жодних аналогів.

2.4 Дослідження роботи сенсорів орієнтації мобільного пристрою

Для визначення положення і орієнтації об'єкта використовується безліч різних сенсорних пристроїв. Найбільш поширеними з цих датчиків є гіроскоп і акселерометр. Хоча подібні за призначенням, вони вимірюють різні речі.

Гіроскоп – це пристрій, який використовує гравітацію Землі, щоб допомогти визначити орієнтацію. Його конструкція складається з вільно обертового диска, що називається ротором, встановленим на осі обертання в центрі більшого і більш стійкого колеса. Коли вісь обертається, ротор залишається стаціонарним, щоб вказувати на центральне тяжіння, і, таким чином, вказувати де знаходиться "низ".

Схематично структура гіроскопа зображена на рисунку 2.4.

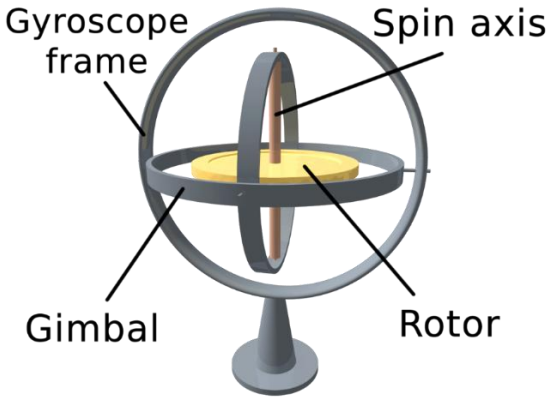

Рисунок 2.4 – Будова гіроскопа

Зазвичай, гіроскоп зроблений шляхом призупинення відносно масивного ротора всередині трьох кілець, що називаються карданним важелем. Монтаж кожного з цих роторів на високоякісних опорних поверхнях гарантує, що на внутрішньому роторі буде дуже малий вплив обертаючого моменту.

Гіроскопи були вперше винайдені і названі у 19 столітті французьким фізиком Жаном-Бернар-Леоном Фуко. Тільки до 1908 року німецький винахідник Х. Аншюц-Кемпфе розробив перший працездатний гірокомпас. Він був створений

для використання в занурювальному пристрої. Потім, в 1909 році, він був використаний для створення першого автопілота.

Першим з мобільних пристроїв, що володіють гіроскопом, став Apple iPhone 4, після чого наявність цієї МЕМС стало мало не обов'язковим вимогам для будь-якого смартфона. Функціональність гіроскопа користувачі змогли оцінювати в багатьох мобільних іграх, де замість одного з двох віртуальних джойстиків з'явилася кнопка пострілу. Ну а цілитися вже доводилося шляхом позиціонування смартфона в просторі, що стало можливо якраз завдяки наявності гіроскопа.

Крім мобільних пристроїв, гіроскопи присутні в контролерах для ігрових приставок PlayStation, Xbox і Wii, де вони функціонують разом з акселерометром. Також ці системи використовуються в камерах з метою оптичної стабілізації для отримання якісних знімків.

Архітектура гіроскопів багато в чому схожа з такою у акселерометрів. Багато з цих пристроїв мають конденсаторну структуру. Такий дизайн, наприклад, використовує в своїх продуктах компанія STMicroelectronics. В основі їх гіроскопа лежить механічний елемент, що працює за принципом камертона і використовує ефект Коріоліса для перетворення кутової швидкості в переміщення чутливої структури. Дві рухливі маси знаходяться в постійному русі, причому в протилежних напрямках. При зміні кутової швидкості починає діяти сила Коріоліса. При цьому напрямок сили Коріоліса перпендикулярно напрямку руху мас. Сила Коріоліса викликає зміщення мас, пропорційне величині кутової швидкості. Оскільки система має конденсаторну структуру, то будь-який зсув викликає зміна електричної ємності. І таким чином кутова швидкість перетвориться в електричний параметр. Тут же варто відзначити, що завдяки використанню спеціальних камертонів гіроскопи STMicroelectronics нечутливі до випадкової вібрації. При такому небажаному впливі на рухливі маси вони обидві будуть зміщуватися в одному напрямку, тим самим не змінюючи ємності конденсатора.

Акселерометр – компактний пристрій, призначений для вимірювання негравітаційного прискорення. Коли об'єкт, в який він інтегрований, переходить

від стану спокою до будь-якої швидкості, акселерометр призначений для реагування на коливання, пов'язані з таким рухом. Він використовує мікроскопічні кристали, які знаходяться під напругою, коли відбуваються вібрації, і з цієї напруги створюється напруга для зчитування прискорення. Акселерометри є важливими компонентами для пристроїв, які відстежують фізичні та інші вимірювання в руху предметів. Перший акселерометр називався машиною Етвуда і був винайдений англійським фізиком Джорджем Етвудом у 1783 році.

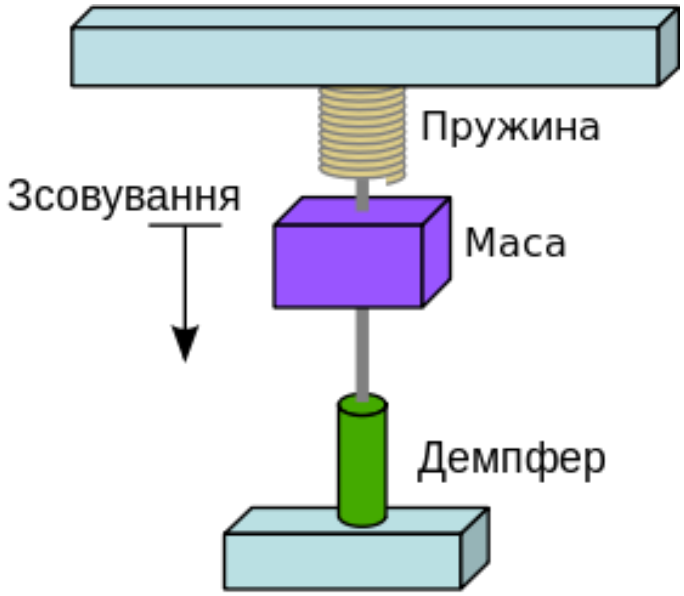

Рисунок 2.5 – Схема найпростішого акселерометру

Існує кілька видів пристроїв в залежності від їх архітектури. Робота акселерометра може грунтуватися на конденсаторному принципі. Рухома частина такої системи є звичайний вантаж, який зміщується в залежності від нахилу пристрою. У міру його зміщення змінюється ємність конденсатора, а саме змінюється напруга. Виходячи з цих даних, можна отримати зміщення грузика, а разом з тим і шукане прискорення.

Найпоширенішим типом акселерометрів є п'єзоелектричні системи. Так само як і в конденсаторних акселерометрах, в їх основі лежить вантаж, який тиском впливає на п'єзокристал. Під тиском він виробляє електричний струм, що дозволяє розрахувати шукане прискорення, знаючи параметри всієї системи.

Існує і ще один тип акселерометрів, який в корені відрізняється від конденсаторного і п'єзоелектричного. Такі акселерометри називаються термальними. Їх архітектура передбачає використання бульбашок повітря. При прискоренні бульбашка відхиляється від свого початкового положення, і це фіксується датчиками. Знаючи, на скільки змістився бульбашка при русі, можна розрахувати величину прискорення.

Основна відмінність між акселерометром та гіроскопом проста: один може відчути обертання, тоді як інший не може. Таким чином, акселерометр може оцінювати орієнтацію стаціонарного елемента по відношенню до поверхні Землі. Але він не в змозі розрізнити, чи об’єкт прискорюється в певному напрямку, чи прискорення здійснюється через гравітаційне тяжіння Землі.

Гіроскоп підтримує свій рівень ефективності, будучи в змозі виміряти швидкість обертання навколо певної осі. При вимірюванні швидкості обертання навколо осі валка літального апарату він визначає фактичне значення, поки об'єкт не стабілізується. Використовуючи ключові принципи кутового моменту, гіроскоп допомагає вказати орієнтацію. Для порівняння, акселерометр вимірює лінійне прискорення на основі вібрації.

Типовий двовісний акселерометр надає користувачам напрямок сили тяжіння в літаку, смартфоні, автомобілі або іншому пристрої. Для порівняння, гіроскоп призначений для визначення кутового положення на основі принципу жорсткості простору. Програми кожного пристрою різко відрізняються, незважаючи на їхнє схоже призначення. Гіроскоп, наприклад, використовується в навігації на безпілотних літальних апаратах, компасах і великих човнах, що в кінцевому підсумку сприяє стабільності в навігації. Акселерометри однаково широко використовуються і можуть бути знайдені в техніці, моніторингу обладнання, будівництві і структурному моніторингу, навігації, транспортуванні і навіть побутовій електроніці.

Поява акселерометра на ринку побутової електроніки, з впровадженням таких поширених пристроїв, як iPhone, що використовується для вбудованого компаса, полегшило його загальну популярність у всіх напрямках програмного

забезпечення. Є декілька основних функцій пов’язаних з акселерометром, які широко використовуються у смартфонах: визначення орієнтації екрана, функціонування у вигляді компасу, скасування дії, просто розхитуючи смартфон [9]. В останні роки його застосування серед споживчої електроніки поширюється і на особисті ноутбуки.

Використання в реальному світі найкраще ілюструє відмінності між даними датчиками. Акселерометри використовуються для визначення прискорення, хоча тривісний акселерометр може ідентифікувати орієнтацію платформи відносно поверхні Землі. Однак, коли ця платформа починає рухатися, її показання стають складнішими для інтерпретації. Наприклад, при вільному падінні акселерометр покаже нульове прискорення. У літаку, що виконує кут нахилу в 60 градусів для повороту, триосьовий акселерометр реєструє 2-G вертикального прискорення, повністю ігноруючи нахил. Зрештою, акселерометр не може бути використаний окремо для підтримки правильної орієнтації літаків.

Натомість акселерометри знаходять застосування у різних споживчих електронних пристроях. Наприклад, серед перших смартфонів, які використовували його, був iPhone 3GS від Apple з введенням таких функцій, як додаток компаса і струс для скасування.

Гіроскоп буде використовуватися в літаку, щоб допомогти в показі швидкості обертання навколо осі обертання літака. Під час нахилу літака, гіроскоп буде вимірювати ненульові значення, поки платформа не стабілізується, після чого буде зчитувати нульове значення, щоб вказати напрямок "вниз". Найкращим прикладом читання гіроскопа є показник висоти на типових літаках. Він представлений кільцевим дисплеєм з розділеним навпіл екраном, верхня половина - блакитним кольором, що вказує на небо, а нижня - червоною для позначення землі.

Заплановане використання кожного пристрою в кінцевому рахунку впливає на їх практичність в кожній використовуваній платформі. Багато пристроїв користуються присутністю обох сенсорів, хоча багато хто покладаються на використання одного. Залежно від типу інформації, яку потрібно зібрати - прискорення або орієнтації - кожен пристрій надасть різні результати [10].

Акселерометр і гіроскоп індивідуально приносять серйозні переваги навігаційній системі; однак обидва пристрої мають невизначеність даних. Обидва з цих датчиків збирають дані про ті ж явища, які є переміщенням об'єкта, злиття вихідних даних для отримання кращого з обох датчиків є хорошим варіантом. Це можна здійснити за допомогою стратегії злиття датчиків [11].

Методи синтезу сенсорів поєднують сенсорні дані з різнорідних джерел і генерують інформацію, яка має меншу невизначеність або більшу точність. Що стосується гіроскопів і акселерометрів, то кожен з них служить для компенсації шуму й помилок дрейфу інших, щоб забезпечити більш повне і точне відстеження руху.

Ця дія поєднання виходів цих датчиків реалізується за допомогою реалізації Калмана або додаткового фільтра [12]. При об'єднанні 3D-акселерометра і даних 3D-гіроскопа найбільш ефективним є співіснування обох функцій в одному пристрої.

Фільтр Калмана є потужним інструментом, який поєднує інформацію при наявності невизначеності (див. рис. 2.6).

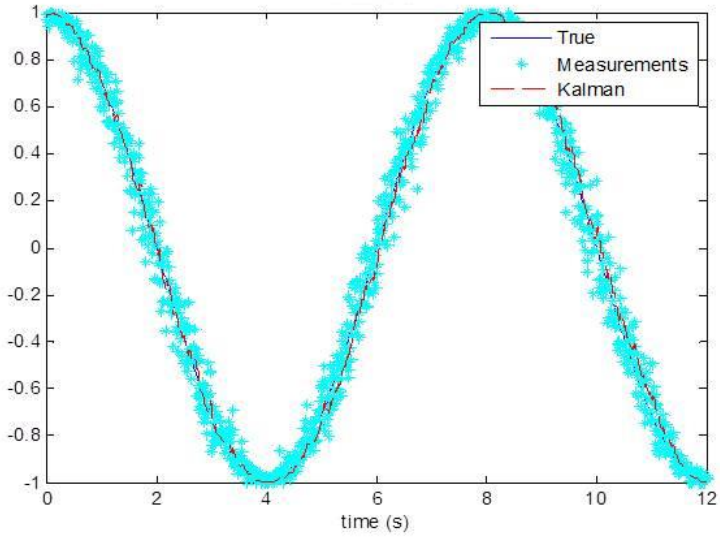

Рисунок 2.6 – Графік прогнозування вимірів

У динамічній системі цей фільтр ідеально підходить для систем, які постійно змінюються. Фільтр Калмана є дискретними, рекурсивними фільтром, що дозволяє

використовувати математичні моделі для отримання оцінки стану системи, незважаючи на наявність значної похибки вимірювань у реальному часі. За допомогою фільтра Калмана, «шумний» акселерометр, гіроскоп і магнітометри можуть бути об'єднані для отримання точного представлення орієнтації і положення.

Фільтр Кальмана в основному складається з двох етапів. На першому етапі для прогнозування стану системи використовується математична модель стану. На наступному етапі це прогнозування стану порівнюється з виміряними значеннями стану. Різниця між прогнозованим і виміряним станом моделюється на підставі оціненого шуму і похибки в системі і вимірах, і виводиться оцінка стану. Оцінку виведення потім використовують у взаємозв'язку з моделлю математичного стану для прогнозування майбутнього стану під час наступного оновлення часу, і цикл починається знову.

2.5 Дослідження роботи Media Capture API та інтеграції з веб-сервером

Захоплення потокового відео і аудіо є пріоритетним напрямком в розробці веб-додатків на протязі багатьох років. Протягом довгого часу для виконання цих завдань використовувалися плагіни Flash і Silverlight.

Але, з часом з'явився HTML5, який надає можливість доступу до апаратних ресурсів. Він створений на основі API JavaScript високого рівня і спирається на апаратні можливості вихідного пристрою.

За останні кілька років з'явилося кілька різних API захоплення мультимедійних файлів. Багато хто усвідомив необхідність доступу до апаратних ресурсів з веб-додатків, і це привело до створення різних специфікацій. Ситуація остаточно заплуталася, і тоді консорціум W3C вирішив сформувати робочу групу. Її єдиним завданням стало наведення порядку у цій сфері. Для цього робоча група

по створенню правил для API пристроїв (DAP) повинна була об'єднати і стандартизувати існуючі пропозиції.

Специфікація захоплення мультимедійних файлів в HTML стала першим стандартом захоплення мультимедійної інформації в веб-додатках, створеним групою DAP. Його робота заснована на перевизначені тега input і додаванні нових значень для параметра accept. Недолік цього API - неможливість додавання ефектів в реальному часі (наприклад, не можна виводити дані з веб-камери на елемент canvas і застосовувати фільтри WebGL). За допомогою захоплення мультимедійних файлів в HTML можна тільки записувати файли мультимедіа або робити знімки.

Вважалося, що в API захоплення мультимедійних файлів в HTML занадто багато обмежень, тому з'явилася нова специфікація, що підтримує будь-які пристрої, включаючи ті, що з'являться в майбутньому. Не дивно, що ця розробка зажадала введення нового елемента device, який став кроком на шляху до створення API getUserMedia(). Однак тег device не був реалізований ні в якому браузері. Елемент device в результаті повністю зник.

Останнім часом спроби створення API для захоплення даних мультимедіа почали приносити відчутні результати - багато в чому завдяки проекту WebRTC (передача даних в Інтернеті в реальному часі). Розробку цієї специфікації курирує спеціальна робоча група W3C WebRTC [13]. В даний час над реалізацією її підтримки в браузерах працюють такі відомі компанії, як Google, Opera, Mozilla, а також ряд інших.

WebRTC - проект, який дозволяє отримувати медіаданні (аудіо та відео) через браузер і встановлювати Peer-to-Peer з'єднання між двома і більше клієнтами, через яке можуть передаватися звичайні дані і медіапотоки. WebRTC заповнює критичний пробіл у веб-платформі. Раніше технології P2P, такі як програми для настільних чатів, могли зробити те, що не вдалося створити в Інтернеті. WebRTC змінює це. WebRTC складається з двох основних компонентів:

– Media Capture and Streams (getUserMedia) – Дозволяє отримати доступ до пристроїв введення, таких як мікрофон і веб-камера. API дозволяє отримати потік медіа з будь-якого з них.

– RTCPeerConnection – за допомогою цих API користувач може відправляти захоплений потік аудіо та відео в режимі реального часу через Інтернет до іншої кінцевої точки WebRTC. За допомогою цих API можна створити з'єднання між локальною машиною і віддаленим вузлом. Він надає способи підключення до віддаленого вузла, підтримання та моніторинг з'єднання та закриття з'єднання після того, як воно більше не потрібно.

– RTCDataChannel – це API дозволяє передавати довільні дані. Кожен канал даних пов'язаний з RTCPeerConnection.

WebRTC має пряме відношення до getUserMedia() функції - першому API в цьому наборі. Ця технологія забезпечує доступ до потокових даних з мікрофона і відеокамери.

```
<video autoplay="true"></video>

<script>
  navigator.getUserMedia({video: true}, function(localMediaStream) {
    var video = document.querySelector('video');
    video.src = window.URL.createObjectURL(localMediaStream);

    video.onloadedmetadata = function(e) {
      // Ready to go. Do some stuff.
    };
  }, function(e) {console.log('Rejected!', e);});
</script>
```

Рисунок 2.7 – Зразок коду для використання Media Capture API

Метод MediaDevices getUserMedia() пропонує користувачеві дозвіл на використання вхідних даних, які створюють MediaStream з треками, що містять потрібні типи носіїв. Цей потік може включати, наприклад, відеотрек (виробляється апаратним або віртуальним джерелом відео, наприклад, камерою, пристроєм запису відео, службою обміну екраном тощо), аудіодоріжкою (аналогічно, виробляється фізичним або віртуальним джерелом звуку, як мікрофон), і, можливо, інші типи доріжок [14].

Для доступу до веб-камері або мікрофону необхідно запросити дозвіл. Перший параметр методу getUserMedia() визначає тип даних, до яких вони вимагають доступ (див. рис. 2.7). Наприклад, щоб звернутися до веб-камери,

необхідно задати для нього значення video. Щоб використовувати одночасно камеру і мікрофон, потрібно вказати video і audio.

Так як цей API може супроводжуватися значними проблемами конфіденційності, getUserMedia() утримується специфікацією для дуже специфічних вимог до сповіщень користувача та керування дозволами. getUserMedia() повинен завжди отримувати дозвіл користувача перед відкриттям будь-якого вхідного засобу для збору інформації, наприклад веб-камери або мікрофона. Браузери можуть пропонувати функцію дозволу "один раз за домен", але веб-сайт повинен запитати принаймні в перший раз, і користувач повинен спеціально надати поточний дозвіл.

Не менш важливими є правила, що стосуються повідомлення. Браузери повинні відображати індикатор, який показує, що камера або мікрофон використовується, над будь-яким апаратним індикатором, який може існувати. Браузер також повиннен показувати індикатор, що надано дозвіл на використання пристрою для введення, навіть якщо пристрій не активно записує на даний момент.

Функція захоплення даних мультимедіа - прекрасний приклад спільної роботи різних API в HTML5. Вона використовується разом з іншими елементами HTML5: audio і video. Замість того щоб вказувати URL файлу мультимедіа, передається URL елемента Blob з об'єкта LocalMediaStream, який представляє веб-камеру.

Також доступний метод MediaDevices.enumerateDevices в цьому API. Цей метод збирає інформацію про пристрої введення / виводу мультимедіа, доступні в системі користувача, такі як мікрофони, камери і динаміки. Ця функція поверне декілька властивостей, включаючи тип (тип) пристрою, наприклад, відеовхід, аудіовхід або аудіовихід. Це дасть змогу надати користувачу можливість вибирати з якої з камер необхідно транслювати відеосигнал.

Peer Connection - це API, який дозволяє встановити Peer-to-Peer з'єднання між різними браузерами.

Нижче представлена спрощена схема з'єднання і взаємодії між двома клієнтами:

– перший клієнт відправляє так званий Offer другого клієнту через сервер (PeerConnection Observer);

– другий клієнт (Remote Peer) відправляє через сервер відповідь першому клієнтові;

– встановлюється P2P з'єднання між клієнтами.

Надалі для роботи такого з'єднання сервер стає опціональним (див. рис. 2.8). Тобто після його виключення дані все також будуть передаватися. Подальша участь PeerConnection Observer'a потрібно для правильного закриття з'єднання, додавання учасників в потік і т.п.

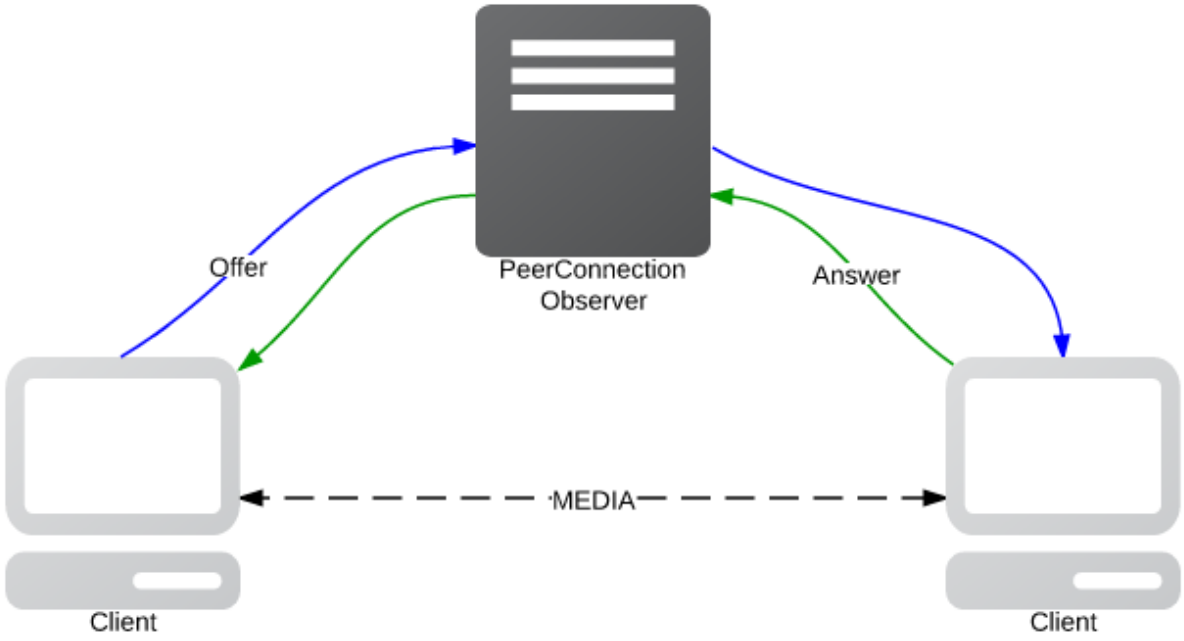

Рисунок 2.8 – Схема встановлення з’єднання між двома клієнтами

Щоб спілкуватися з іншим вузлом через веб-браузер, веб-переглядач кожної людини повинен виконувати такі дії:

– погодитися на початок з’єднання;

– дізнатися, як знайти один одного;

– обхід захисту та захисту брандмауера;

– передавати всі мультимедійні комунікації в режимі реального часу.

Одним з найбільших викликів, пов'язаних із зв'язком однорангових вузлів на основі браузера, є знання того, як знайти і встановити з'єднання з мережним сокетом з іншим веб-браузером, щоб двонаправлено передавати дані. Звичайне HTTP з’єднання не підтримує такої можливості.

Коли веб-додаток потребує даних або ресурсу, він вибирає його з певного сервера. Проте, якщо ви хочете створити одноранговий відеочат, підключившись безпосередньо до чужого веб-переглядача, ви не знаєте адресу, оскільки інший веб-переглядач не є відомим веб-сервером. Отже, для встановлення з'єднання p2p потрібно щось більше.

Комп'ютер користувача зазвичай не має статичної публічної IP-адреси. Причина полягає в тому, що комп'ютер розміщений за брандмауером і пристроєм перекладу доступу до мережі (NAT).

Пристрій NAT - це пристрій, який переводить приватні IP-адреси зсередини брандмауера на загальнодоступні IP-адреси. Пристрої NAT необхідні для захисту та обмежень IPv4 на доступні загальнодоступні IP-адреси. Саме тому веб-програма не повинна припускати, що поточний пристрій має загальнодоступну статичну IP-адресу. Враховуючи участь NAT пристрою, ваш браузер повинен з'ясувати IP-адресу машини, яка має браузер, з яким ви хочете спілкуватися [15].

Кожен учасник спочатку встановлює IP-адресу публічного доступу. Потім "канали" сигналізації даних динамічно створюються для виявлення однорангових вузлів і підтримки однорангових переговорів і встановлення сеансів між користувачами.

Компонент RTCPeerConnection надає веб-розробникам абстракцію від складності з боку структури внутрішніх елементів системи. Кодеки та протоколи, що використовуються WebRTC, роблять величезну роботу, щоб зробити взаємодію в реальному часі можливим, навіть через ненадійні мережі:

- приховування втрат пакетів;
- скасування відлуння;
- адаптивність смуги пропускання;
- динамічне буферування джиттера;

– автоматичне регулювання посилення;

– зменшення шуму;

– зображення "очищення".

2.6 Опис архітектурної моделі

Після аналізу предметної галузі і доступних аналогів була розроблена загальна схема взаємодії усіх компонентів системи між собою [16], дана схема представлена на рисунку 2.9.

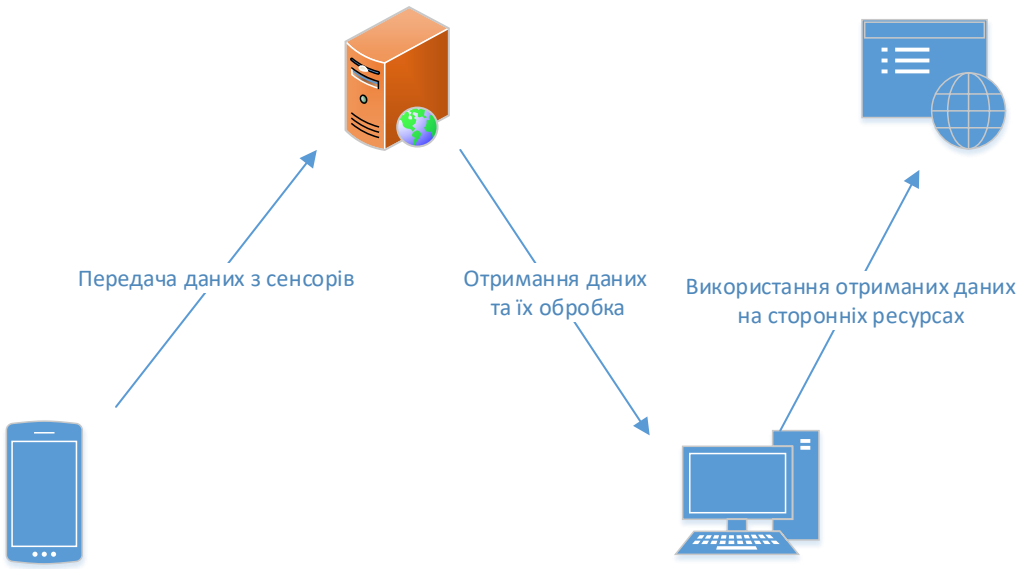

Рисунок 2.9 – Схема взаємодії смартфону з персональним комп’ютером

Отже для реалізації усіх поставлених задач розроблюваний додаток має складатися з трьох компонентів:

– Веб-клієнт. Виступає у якості пристрою для зчитування усіх показників смартфону і передає ці данні по безпечному каналу до веб-серверу.

– Веб-сервер. Виступає у якості з’єднуючої ланки між смартфоном та комп’ютером і також регулює з’єднання між ними.

– Десктопний додаток. Виступає у якості отримувача усіх даних із веб-серверу і потім оброблені дані передає операційній системі для подальшого їх використання.

Веб-клієнт буде розроблений з використанням JavaScript мови програмування. А веб-сервер та десктопний додаток за допомогою мови програмування C#.

C # відноситься до сім'ї мов з C-подібним синтаксисом, з них його синтаксис найбільш близький до C++ і Java. Мова має статичну типізацію, підтримує поліморфізм, перевантаження операторів (в тому числі операторів явного і неявного приведення типу), делегати, атрибути, події, властивості, узагальнені типи і методи, ітератори, анонімні функції з підтримкою замикань, LINQ, виключення, коментарі у форматі XML [17].

В якості інтегрованого середовища розробки (IDE) було обрано Microsoft Visual Studio. Microsoft Visual Studio - це набір інструментів для створення програмного забезпечення: від планування до розробки інтерфейсу користувача, написання коду, тестування, налагодження, аналізу якості коду і продуктивності, розгортання в середовищах клієнтів і збору даних телеметрії по використанню. Ці інструменти призначені для максимально ефективної спільної роботи [18].

3 ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

3.1 UML проектування програмної системи

Перед початком створення програмного засобу необхідно спроектувати основні частини програми. Це дасть змогу чітко зрозуміти які компоненти будуть розроблятися і який архітектурний стиль вибрати для реалізації системи. Наглядні діаграми полегшують сприйняття та розуміння поставленої задачі для розробника та дають змогу для розробника і замовника впевнитися, що вони обидва правильно зрозуміли одне одного [19].

У якості мови для написання діаграм було обрано UML. Мова уніфікованого моделювання (UML) – це стандартизована мова моделювання, що дозволяє розробникам вказувати, візуалізувати, конструювати та документувати артефакти програмної системи. Таким чином, UML робить ці артефакти масштабованими, безпечними та надійними у виконанні. UML є важливим аспектом, що бере участь у розробці об'єктно-орієнтованого програмного забезпечення. Він використовує графічні позначення для створення візуальних моделей програмних систем.

Архітектура UML заснована на мета-об'єктах, які визначають основу для створення мови моделювання. Вони досить точні, щоб відтворити всю програму. Будь-які UML діаграми можуть бути розроблені на різних платформах з використанням різних технологій і можуть використовуватися з усіма процесами протягом всього циклу розробки програмного забезпечення.

UML розроблено, щоб дозволити користувачам розробити зрозумілу, готову до використання мову візуального моделювання. Крім того, він підтримує такі високорівневі концепції розвитку, як фреймворки та шаблони проектування.

Для побудови діаграм було використано додаток Microsoft Visio 2013.

Діаграма послідовностей показує впорядкований процес взаємодії певних об’єктів між собою у певний проміжок часу. На цій діаграмі відображають ті об’єкти, які беруть участь у даному процесі. Основною метою цієї діаграми є показити динаміку проходження будь-якого процесу.

Ця діаграма зображена на рисунку 3.1. Вона відображає в цілому як відбувається більшість взаємодій користувача з системою. Для початку користувач посилає запит на сервер для підтвердження підключення клієнту мобільного пристрою до PC, яким користувач бажає керувати. Якщо з'єднання успішно встановлено, то клієнт має змогу передавати дані з сенсорів до веб-серверу. Після цього API перевіряє на коректність передані дані і в разі їх невірності одразу повертає клієнту відповідь з кодом помилки 400. Якщо дані вірні, то сервер передає їх далі до бізнес логіки, яка їх обробляє, робить необхідні операції і потім отриманий результат повертається до клієнта.

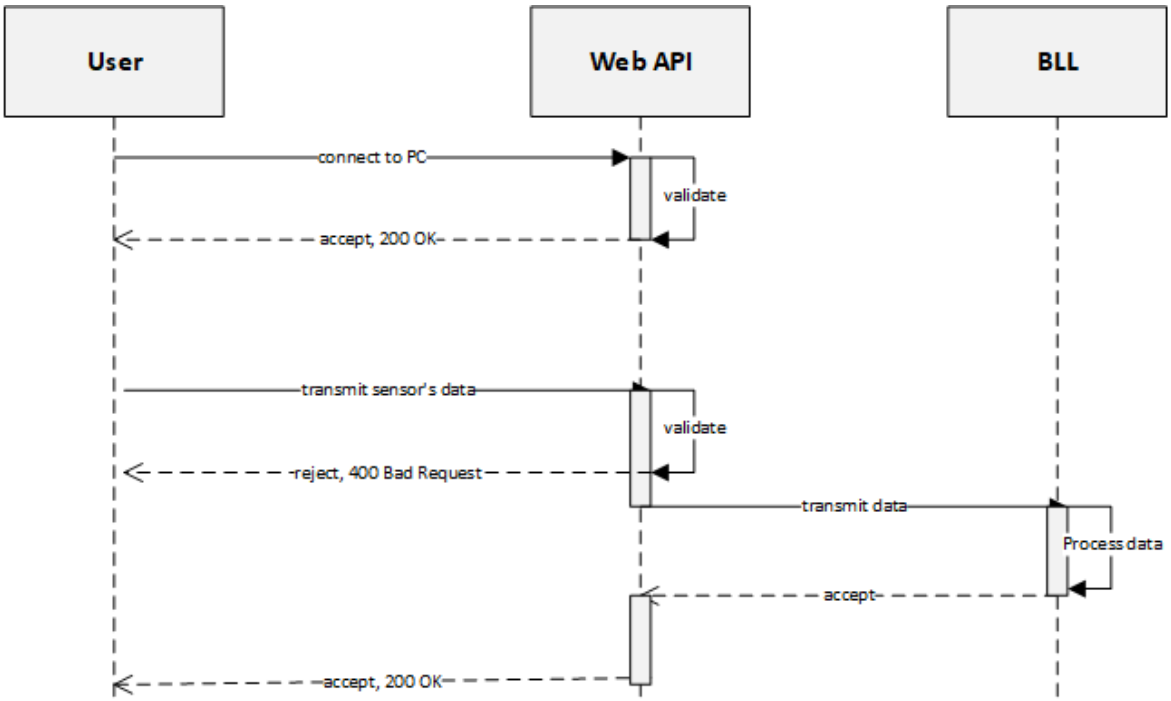

Рисунок 3.1 – Діаграма послідовностей

На рисунку 3.2 представлена діаграма розгортання. Вона використовується для відображення основних компонентів системи та їх розміщення по окремих серверах. Як видно з діаграми дане програмне забезпечення розділено на три основних компоненти: веб клієнт (мобільний пристрій), веб сервер та персональний комп'ютер користувача, яким здійснюється керування. Взаємодія між клієнтом та сервером відбувається за допомогою http/https протоколу. Також на діаграмі видно, що сервер складається з двох основних компонентів, які

реалізовані таким чином, що можуть бути розділені на різні вузли. У разі помилки всі дані про неї записуються у Log файл. Результати обробки усіх даних передаються до PC, яким здійснюється керування за допомогою мобільного пристрою, де за допомогою доступу до API самої ОС користувача здійснюється керування необхідними пристроями комп'ютера.

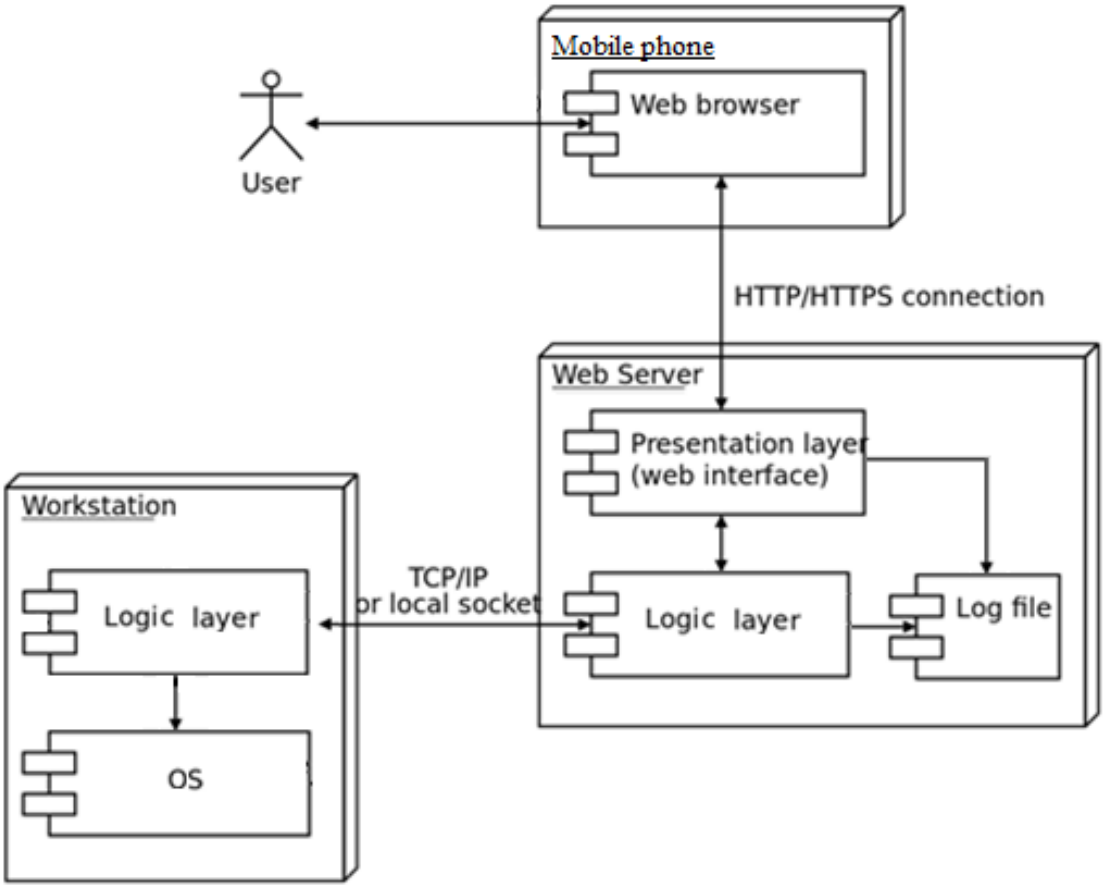

Рисунок 3.2 – Діаграма розгортання

Таким чином стає зрозуміло, що програмна система буде складатися з трьох компонентів, кожний з яких буде розташований на окремих пристроях: смартфон, основний сервер, персональній комп'ютер користувача.

3.2 Архітектура програмної системи

Архітектура програмного забезпечення відіграє дуже велику роль у всій системі. Саме проектування правильної архітектури системи дозволить системі розвиватися у майбутньому. Адже будь-яка дрібна помилка на початку проектування системи може суттєво погіршити працездатність системи та навіть зовсім загальмувати роботу системи. І вирішити такі помилки можливо буде лише повність переписавши усю систему з використанням нової архітектури, що може зайняти дуже велику кількість часу.

Для реалізації сервісу вирішено використовувати клієнт-серверну архітектуру. Ця архітектура є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережевих додатків і передбачає взаємодію та обмін даними між ними. Саме тому для реалізації цієї архітектури було вирішено розбити серверну частину на два рівні:

– Presentation layer – цей рівень відповідає за побудову HTML сторінки для користувача, за отримання усіх вхідних запитів від клієнтів та подальшу передачу їх до наступного рівню для обробки та за авторизацію користувача (підключення користувача до свого комп’ютера та підтримка відкритого з’єднання між пристроями).

– Business layer (рівень бізнес-логіки) – даний рівень відповідає за обробку усіх запитів та подальшу переадресацію даних із сенсорів смартфону на необхідний пристрій.

На етапі розробки системи немає жодної інформації для довгострокового зберігання, тому програмна система може бути розроблена без рівню доступу до даних (Data Access layer) і без використання бази даних. Але враховуючи дану архітектуру новий рівень може бути з легкістю доданий до програмної системи, без повної її переробки.

Під час реалізації цих шарів було винесено окремо інтерфейси для кожного з шару у окремий проект. Це дозволило зменшити кількість залежностей між компонентами системи.

Веб-інтерфейс користується інтерфейсами сервісів з бізнес-логіки, але він нічого не знає про реальну реалізацію. Це архітектурне рішення дає дуже велику гнучкість системі. Вона може бути з легкістю розбита на окремі рівні, які можуть знаходитися на різних серверах, наприклад з використанням WCF сервісів, що дасть змогу розподілити навантаження на систему і зробити її більш стійкою. Також такий підхід дає можливість писати юніт-тести на кожний метод сервісу, тестуючи лише один компонент.

Для даної програмної системи сервер виступає у якості центрального вузлу, що зв’язує комп’ютер та мобільний пристрій користувача.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

4.1 Вибір засобів розробки

В результаті проектування та моделювання програмної системи було визначено, що додаток має складатися із трьох компонентів: мобільний веб-клієнт, веб-сервер та десктопний додаток. Так як у даній версії програми не має необхідності в довгостроковому зберіганні даних, то даний програмний продукт буде розроблений без використання будь-якої бази даних.

Для розробки веб-серверу буде використана технологія для створенная Web-додатків мовою C# з використанням ASP.NET MVC 5 та Web API фреймворків. Основною особливістю цього веб-серверу буде розробка можливості взаємодії серверу з клієнтами в обох напрямках. Такі можливості надає використання технології SignalR.

ASP.NET SignalR - це бібліотека для розробників ASP.NET, що спрощує процес додавання веб-функціональних можливостей у реальному часі до додатків. Веб-функціональність у режимі реального часу - це можливість мати серверний код, який передає контент для підключених клієнтів, як тільки він стане доступним, замість того, щоб сервер чекав, коли клієнт запитає нові дані. SignalR може використовуватися для додавання будь-якого типу "реального часу" веб-функціональності до програми ASP.NET. Хоча чат часто використовується як приклад, з використанням SignalR можливо зробити набагато більше [20]. Кожного разу, коли користувач оновлює веб-сторінку, щоб побачити нові дані, або сторінка впроваджує довгі опитування для отримання нових даних, вона є кандидатом для використання SignalR. Приклади включають приладові панелі та програми моніторингу, спільні програми (наприклад, одночасне редагування документів), оновлення прогресу роботи та форми в реальному часі. SignalR також дозволяє повністю нові типи веб-додатків, які вимагають оновлення високої частоти з сервера, наприклад, ігри реального часу. SignalR надає простий API для створення віддалених викликів процедур від сервера до клієнта (RPC), які викликають функції

JavaScript у браузерах клієнтів (та інших клієнтських платформах) з коду сервера .NET (див. рис. 4.1). SignalR також включає API для керування з'єднаннями (наприклад, підключення і відключення подій) і групування з'єднань.

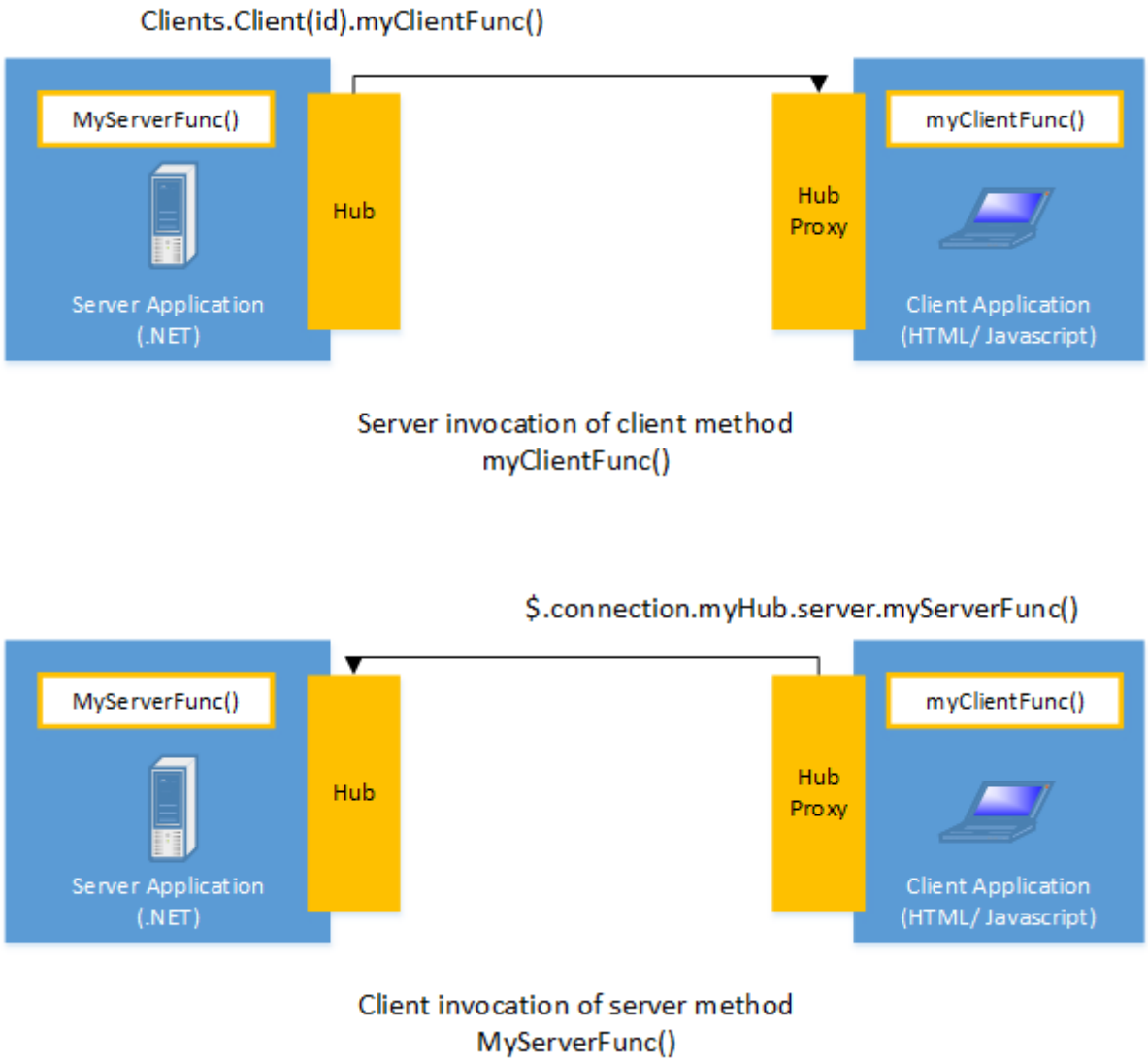

Рисунок 4.1 – Приклад взаємодії клієнту і серверу за допомогою SignalR

SignalR керує з'єднаннями автоматично, і дозволяє передавати повідомлення всім підключеним клієнтам одночасно. Також можна надсилати повідомлення певним клієнтам. Зв'язок між клієнтом і сервером є стійким, на відміну від класичного HTTP-з'єднання, яке повторно встановлюється для кожного зв'язку.

SignalR підтримує функціональність "server push", в якій код сервера може викликати код клієнта в браузері за допомогою віддалених викликів процедур (RPC), а не у вигляді запиту-відповіді, загальноприйнятого в Інтернеті.

Застосування SignalR дозволяє масштабувати до тисяч клієнтів за допомогою Service Bus, SQL Server або Redis. SignalR розробляється з відкритим вихідним кодом, доступним через GitHub.

SignalR – це абстракція над деякими транспортними засобами, необхідними для роботи в режимі реального часу між клієнтом і сервером. З'єднання SignalR починається як HTTP, а потім змінюється до з'єднання WebSocket, якщо воно доступне. WebSocket є ідеальним механізмом для SignalR, оскільки він робить найбільш ефективне використання пам'яті сервера, має найнижчу затримку і має основні функції (наприклад, повне дуплексне з’єднання між клієнтом і сервером), але він також має найсуворіші вимоги: WebSocket вимагає, щоб сервер використовував Windows Server 2012 або Windows 8 і .NET Framework 4.5. Якщо ці вимоги не виконуються, SignalR намагатиметься використовувати інші транспортні засоби для здійснення своїх з'єднань. Окрім з’єднання за допомогою технології WebSocket, SignalR може вибирати інші найбільш підходящі методи для взаємодії між клієнтом і сервером:

– Server Sent Events, також відомий як EventSource (якщо веб-переглядач підтримує Server Sent Events, в основному всі браузери, окрім Internet Explorer.)

– Forever Frame (лише для Internet Explorer). Forever Frame створює прихований IFrame, який робить запит до сервера, який не завершується. Потім сервер постійно надсилає скрипт клієнту, який негайно виконується, забезпечуючи одностороннє реальне з'єднання від сервера до клієнта. Підключення від клієнта до сервера використовує окреме з'єднання від сервера до клієнта, і, як і стандартний HTTP-запит, для кожного фрагмента даних, який потрібно надіслати, створюється нове з'єднання.

– Ajax long polling. Довге опитування не створює постійного з'єднання, але замість цього запитує сервер із запитом, який залишається відкритим, доки сервер не відповість, і в цей момент з'єднання закривається, і негайно відкривається

нове з'єднання. Це може призвести до деякої затримки під час створення нового з'єднання.

API SignalR містить дві моделі для обміну даними між клієнтами та серверами: Persistent Connections та Hubs.

З'єднання являє собою просту кінцеву точку для відправлення одного одержувача, згрупованих або широкомовних повідомлень. Persistent Connections API (представлений у коді .NET класом PersistentConnection) надає розробнику прямий доступ до протоколу комунікації низького рівня, який надає SignalR. Використання моделі зв'язку Connections схоже з API на основі з'єднання, наприклад, Windows Communication Foundation.

Hab - це більш високорівневий конвеєр, який побудований на Connection API, що дозволяє клієнту і серверу безпосередньо викликати методи один одного безпосередньо. SignalR обробляє диспетчеризацію через межі машини, як за допомогою магії, дозволяючи клієнтам викликати методи на сервері так само легко, як і локальні методи, і навпаки. Використання комунікаційної моделі Hubs дуже схожі з використанням API віддаленого виклику, наприклад .NET Remoting. Використання Hub також дозволяє передавати сильно типізовані параметри методам, що дозволяє прив'язувати моделі.

Для реалізації мобільного веб-клієнту було вирішено використовувати ASP.NET MVC фреймворк, що дозволяє отримувати та відправляти різноформатні дані по протоколу HTTP. Веб-клієнт представляє собою набір простих веб-сторінок, розроблених за допомогою HTML та CSS, кожна з яких надає користувачу можливість взаємодіяти зі своїм комп’ютером за допомогою обраних датчиків та сенсорів з використанням JavaScript Device API. Так як кожна сторінка не має складної логіки взаємодіє з користувачем або різноманітного динамічного контенту, тому для швидкості відображення та легкості усіх сторінок було обрано використання звичайного JavaScript, замість потужних та важких фреймворків для UI, таких як Angular, Knockout та ін.

Так як для використання повного функціоналу програмної системи необхідна підтримка нових функцій JavaScript Device API, тому даний сайт може

функціонувати тільки на останніх версіях таких мобільних браузеріва як Google Chrome та Firefox. Окрім цього, браузери можуть взаємодіяти з датчиками мобільного пристрою лише у випадку HTTPS з’єднання з сервером.

І для розробки десктопного додадтку було обрано використання Windows Forms через свою простоту та швидкість розробки додатку. Даний програмний продукт не може використовувати веб-сайт на стороні комп’ютера через політику безпеки операційних систем, адже веб-сайти через браузер не мають можливості керувати пристроями ОС та створювати свої віртуальні пристрої. Тому було вирішено розробити простий UI клієнт для комп’ютеру, який буде проміжною ланкою між даними, отриманими з сенсорів мобільного пристрою та операційною системою, яка буде отримувати дані та обробляти їх.

Так як однією з функцій програмної системи є транслювання відео з мобільного пристрою у якості веб-камери самого комп’ютеру, то виникає необхідність у розробці своєї віртуальної камери, яка би мала можливість транслювати отриманий сигнал у відео потік комп’ютера. Віртуальна веб-камера, як правило, є лише програмною реалізацією, яку програма виявляє як пристрій з фізичним поданням. Зазначені програми використовують API для роботи з веб-камерами, а можливість розширювати API і додавати власне джерело відео - це спосіб створення віртуальної веб-камери.

У Windows є кілька API для споживання відео джерел: Video for Windows, DirectShow, Media Foundation (в хронологічному порядку реалізації даних технологій).

Video for Windows є не досить розширюваним і обмеженим у загальних можливостях. Дана технологія дозволить створити віртуальний пристрій, якщо створити драйвер режиму ядра для віртуальної камери. Video for Windows було вперше представлено в листопаді 1992 року. Дана технологія була розроблена як реакція на технологію QuickTime від Apple Computer, яка додала цифрове відео на платформу Macintosh. Версія для перегляду лише відеозаписів також була доступна як безкоштовний додаток до Windows 3.1 і Windows 3.11; вона стала невід'ємною складовою Windows 95 і пізніших версій. Як і QuickTime, Video for Windows мало

три ключові аспекти: Audio Video Interleave (AVI), формат файлу контейнера, призначений для зберігання цифрового відео; інтерфейс прикладного програмування (API), який дозволив розробникам програмного забезпечення запускати або маніпулювати цифровим відео в своїх власних додатках; набір програмного забезпечення для відтворення та редагування цифрового відео. Video for Windows було замінено на випуск ActiveMovie липня 1996 року, пізніше відомий як DirectShow. Вперше він був випущений як бета-версія разом з другою бета-версією Internet Explorer 3. ActiveMovie був випущений у вигляді безкоштовного завантаження, або окремо, або в комплекті з Internet Explorer. Проте ActiveMovie не підтримував захоплення відео. Video for Windows все ще використовувалося для захоплення відео до випуску драйверів для захоплення відео, які тільки почали ставати популярними в 2000 році.

DirectShow - це API, який використовується більшістю відеозаписних додатків Windows, і він присутній у всіх версіях Windows, включаючи Windows 10 (окрім Windows RT). Ця технологія цілком розширювана і в більшості випадків термін "віртуальна веб-камера" відноситься до віртуальної веб-камери DirectShow.

Media Foundation є можливим наступником DirectShow, але його можливості захоплення відео в частині розширюваності просто не існує. Microsoft вирішила не дозволяти користувальницькі джерела відеододатків, які могли б працювати так само, як веб-камери. Завдяки складності Media Foundation, його використовує невелика кількість додатків. Для реалізації віртуальної веб-камери для програми Media Foundation необхідно знову, як і у випадку з Video for Windows, реалізувати драйвер режиму ядра. Тому, спираючись на дану інформацію було вирішено використовувати DirectShow для розробки віртуальної веб-камери.

Microsoft DirectShow - це архітектура для потокового мультимедіа на платформі Microsoft Windows. DirectShow забезпечує якісну зйомку і відтворення мультимедійних потоків. Він підтримує широкий спектр форматів, включаючи Advanced Systems Format (ASF), Motion Picture Experts Group (MPEG), Audio-Video Interleaved (AVI), MPEG Audio Layer-3 (MP3) і WAV-файли. Він підтримує запис з цифрових та аналогових пристроїв на основі моделі драйверів Windows (WDM) або

відео для Windows. Він автоматично виявляє та використовує апаратні засоби прискорення відео та аудіо, якщо вони є, але також підтримує системи без апаратного прискорення.

DirectShow базується на компонентній об'єктній моделі (COM). Щоб написати програму або компонент DirectShow, розробнику необхідно розуміти програмування COM-клієнтів. Для більшості програм не потрібно реалізовувати власні об'єкти COM. DirectShow надає необхідні компоненти. Але якщо виникає необхідність розширити DirectShow, написавши власні компоненти, то необхідно реалізувати їх як об'єкти COM. DirectShow призначений для C++. Microsoft не надає керований API для DirectShow. Тому використання .NET Framework надасть можливість інтегрувати зовнішні бібліотеки, написані на іншій мові, до спільного додатку, написаного з використанням C#.

DirectShow спрощує відтворення мультимедіа, перетворення формату та захоплення відео. У той же час, він надає доступ до базової архітектури керування потоком для додатків, які потребують спеціальних рішень. Також надає можливість створювати власні компоненти DirectShow для підтримки нових форматів або спеціальних ефектів.

Робота з мультимедіа представляє кілька основних проблем:

– мультимедійні потоки містять велику кількість даних, які необхідно обробляти дуже швидко;

– аудіо та відео повинні бути синхронізовані, щоб вони запускалися і зупинялися одночасно, і відтворювався з однаковою швидкістю;

– дані можуть надходити з багатьох джерел, включаючи локальні файли, комп'ютерні мережі, телевізійні передачі та відеокамери;

– дані надходять у різноманітних форматах, таких як Audio-Video Interleaved (AVI), Advanced Streaming Format (ASF), Motion Picture Experts Group (MPEG), and Digital Video (DV);

– програміст не знає заздалегідь, які апаратні пристрої будуть присутні в системі кінцевого користувача.

DirectShow призначений для вирішення кожного з цих завдань. Його основна мета - спростити завдання створення цифрових мультимедійних додатків на платформі Windows, ізолюючи програми від складності перенесення даних, апаратних відмінностей і синхронізації.

Для досягнення пропускної здатності, необхідної для потокового відео та аудіо, DirectShow використовує Direct3D і DirectSound, коли це можливо. Ці технології ефективно передають дані до звукових і графічних карт користувача. DirectShow синхронізує відтворення за допомогою інкапсулювання медіаданих у зразках, позначених часом. Для обробки різноманітних джерел, форматів і можливих пристроїв, DirectShow використовує модульну архітектуру, в якій додаток змішується і відповідає різним компонентам програмного забезпечення, що називаються фільтрами.

DirectShow надає фільтри, які підтримують пристрої захоплення та налаштування на основі моделі драйверів Windows (WDM), а також фільтри, що підтримують старі карти захоплення Video for Windows (VfW), і кодеки, написані для диспетчера стиснення звуку (ACM) і диспетчера стиснення відео (VCM) інтерфейсів.

4.2 Опис програмної системи

Як вже було розглянуто раніше, то вся програмна система буде складатися з трьох основних компонентів: веб-клієнта, веб-сервера та десктопного додатку.

Веб-сервер реалізован у вігляді REST-сервісу, за допомогою якого веб-клієнт та додаток на комп’ютері зможуть взаємодіяти між собою та використовувати функції системи. Дані між вказанами компонентами будуть передаватися у форматі JSON.

Однією з основних цілей проектування і розробки даної системи було створити максимально простий та зручний інтерфейс, який би не потребував у

користувача додаткових знань по роботі з комп'ютером. Тому увесь інтерфейс відповідає принципам юзабіліті та потребує від користувача мінімальну кількість дій аби працювати з даним додатком.

Десктопний клієнт складається з одного вікна за допомогою якого користувач може підключитися до серверу для подальшої роботи з додатком, або відключитися, коли додаток більше не потрібен (див. рис. 4.2). Після встановлення з'єднання з сервером створюється секретний ключ, за допомогою якого користувач може приєднатися зі свого мобільного пристрою до комп'ютера.

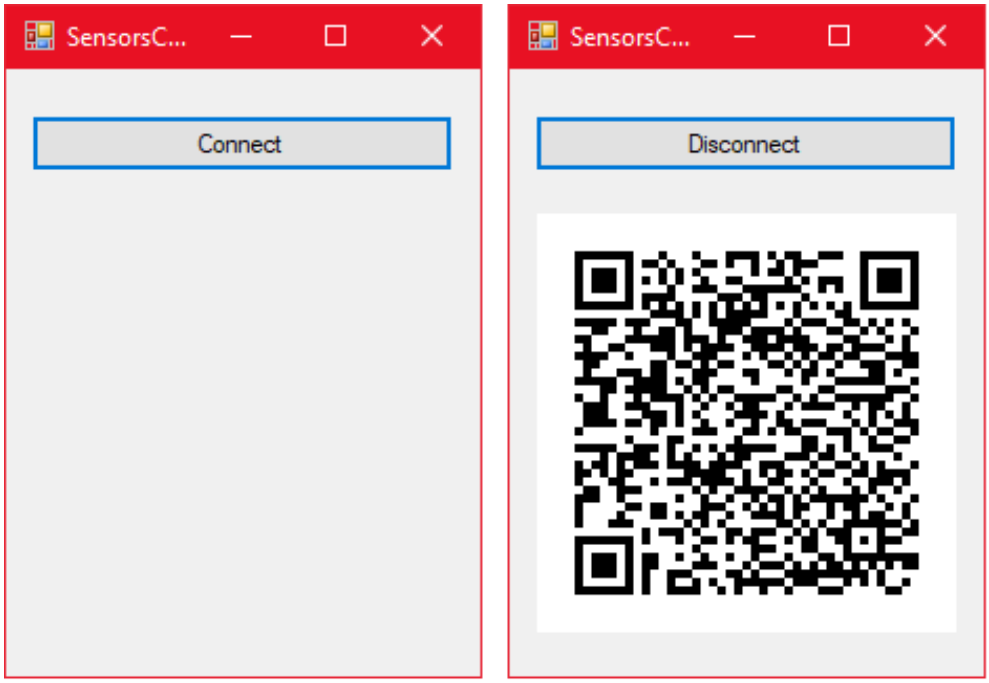

Рисунок 4.2 – Інтерфейс основної програми для керування комп'ютером

Згенерований секретний ключ відображається у вигляді QR коду, який необхідно зчитати з мобільного пристрою. На цьому налаштування з'єднання між комп'ютером та смартфоном завершується і програмна система готова для подальшої роботи. Даний секретний ключ створюється кожного разу новий і передається по закодованому з'єднанню, тому у зловмисників не має можливості перехопити його та отримати доступ до керування пристроєм користувачів даної системи.

Веб-клієнт для користувача також має простий та зрозумілий інтерфейс і складається з декількох сторінок. Також даний сайт розробляється в основному для мобільних пристроїв, тому весь контент на ньому одразу оптимізован для відображення на екранах смартфонів. При першому вході до сайту користувачу буде запропановано зчитати QR код для того, щоб підключитися до системи і отримати можливість користуватися даним програмним продуктом.

Після авторизації, користувач потрапляє на головну сторінку (див. рис. 4.3), де він матиме змогу обрати необхідні йому функції або завершити з’єднання.

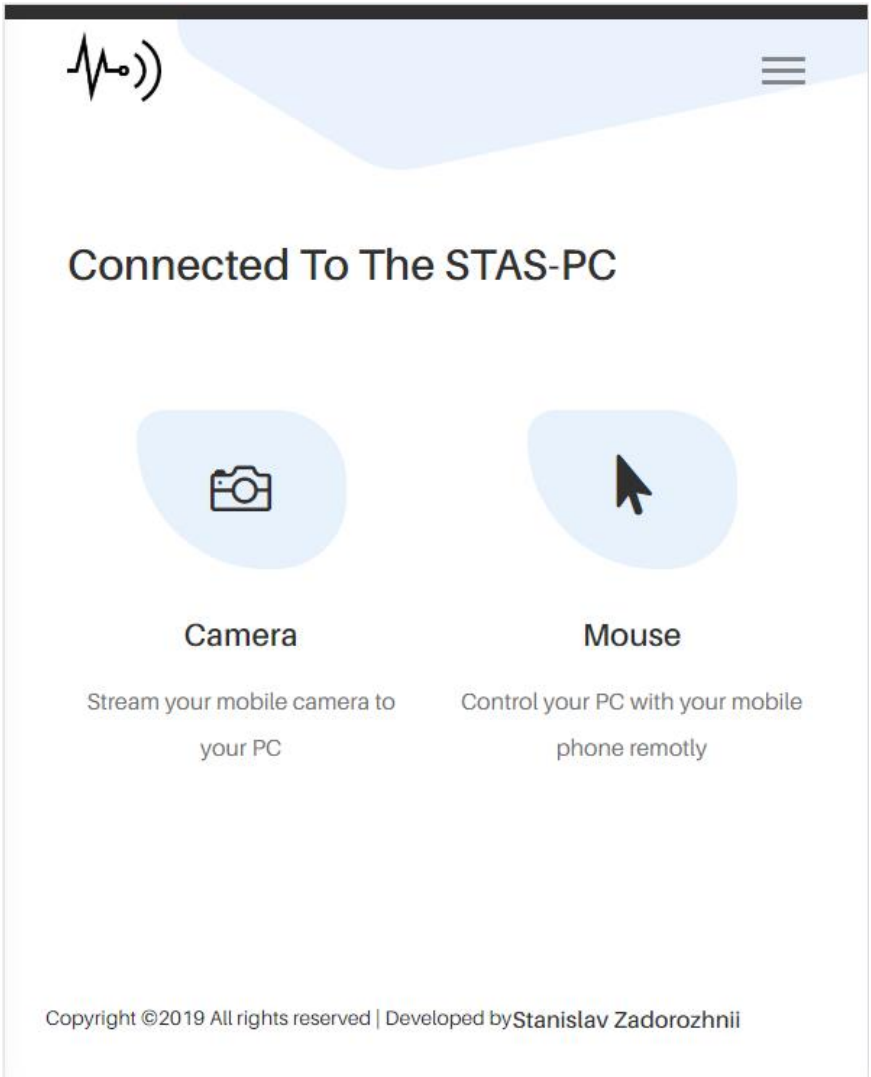

Рисунок 4.3 – Головна сторінка програми

Також на головній сторінці користувач має можливість переглянути додаткову інформацію про комп’ютер, до якого він підключен.

З головної сторінки можливо перейти до будь-якої функції, які доступні в даній програмній системі.

Після вибору опції “Mouse” користувач буде перенаправлений до сторінки, яка надаю можливість керувати мишкою компьютера за допомогою жестів, акселерометра та гіроскопа (див. рис. 4.4).

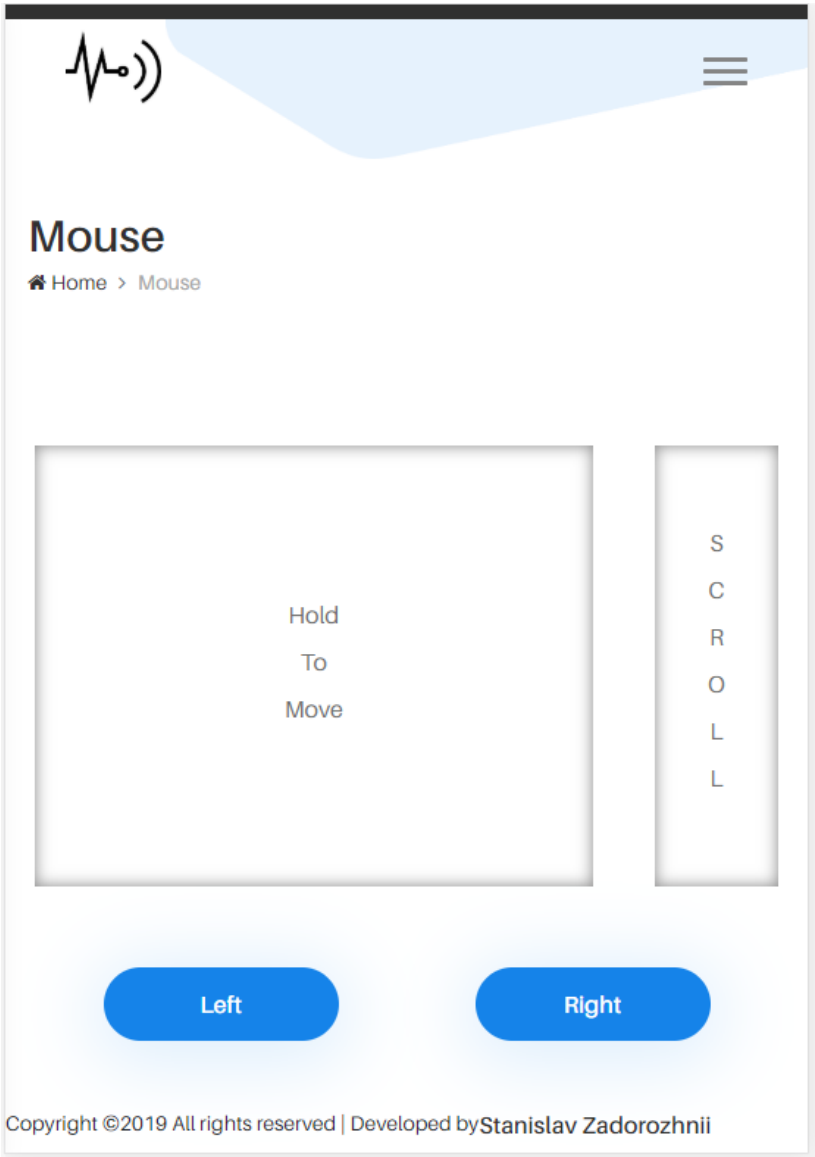

Рисунок 4.4 – Сторінка для керування мишкою

За допомогою даної сторінки можливо керувати курсором на комп’ютері. Для цього необхідно утримавати кнопку “Hold to Move”, після цього активуються датчики смартфону і дані з них передаються до комп’ютера, де враховуючи амплітуду і напрям руху смартфону переміщується курсор миші. Також за

допомогою жестів у спеціальній секції можно прокручувати сторінку на комп’ютері вгору або вниз. І на даній сторінці присутні кнопки, які емулюють натискання лівої і правої кнопки миші.

Отже, за допомогою даної сторінки користувач може віддалено керувати своїм комп’ютером. Наприклад, даний функціонал може бути дуже корисний при демнострації презентацій.

Наступна функція, яка доступна користувачам даної системи – це використання камери телефону у якості веб-камери комп’ютера (див. рис. 4.5).

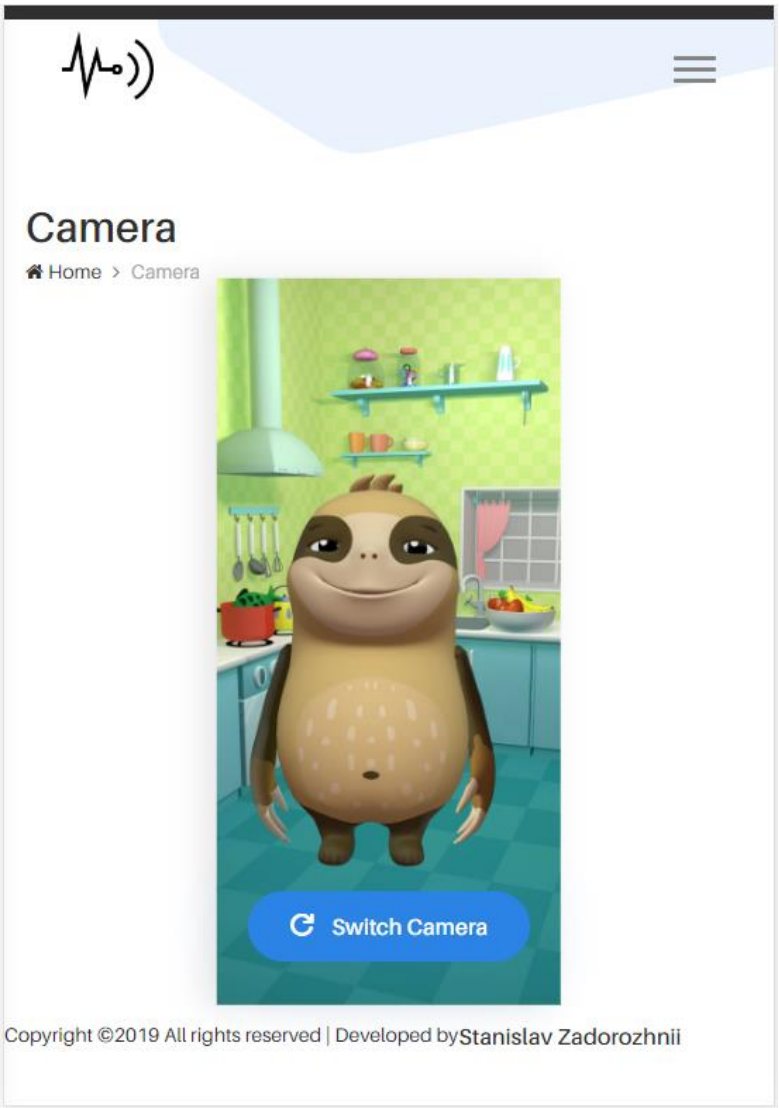

Рисунок 4.5 – Сторінка для трансювання відео

Після вибору опції “Camera” користувач буде перенаправлений до сторінки де він зможе трансювати відео з камери на свій комп’ютер.

Дана сторінка складається з двох елементів: блок із зображенням з камери та кнопка для зміни камери, яка використовується для трансляції відео. JavaScript Device API надає можливість отримувати список всіх доступних відео потоків, які можливо використовувати для трансляції відео. Між цими ресурсами можливо перемекатися за допомогою кнопки “Switch”, яка знаходится під відео користувача.

При першому зверненні до цієї сторінки браузер запитає дозвіл для використання камери стороннім сайтом. У разі надання користувачем дозволу сигнал з камери почне транслюватися на комп’ютер у вигляді зовнішньої веб-камери.

У разі необхідності користувач може вийти із системи та відключитися від комп’ютера за допомогою відповідного пункту меню “Disconnect”.

У результаті розробки програмної системи було отримано додаток, який задовольняє усім критеріям запропонованим у ході дослідження предметної галузі. Даний програмний продукт надає можливість керувати комп’ютером по безпечному каналу зв’язку, також він економить кошти користувачів на придбанні додаткових зовнішніх пристроїв для ПК. Однак, швидкість передачі сигналу є меншою у порівнянні з зовнішніми пристроями, тому можна побачити невеликі затримки при використанні відео зв’язку, проте така затримка є не суттєвою і не впливає на роботу з даною програмною системою.

ВИСНОВКИ

В процесі роботи був проведений аналіз предметної галузі, були виявлені доступні API, які може використовувати веб-браузер. На основі отриманих даних було обрано два API за допомогою, яких можна побудувати додаток, який би взаємодіяв з персональним комп’ютером з використанням обраних сенсорів:

– Device Motion and Orientation API;

– Media Capture and Streams API.

У ході роботи було проаналізовано існуючі аналоги, які мають схожий функціонал по взаємодії смартфону з комп’ютером. А також було порівняно функціонал додатку з існуючими фізичними приладами, які зараз виконують необхідні функції.

Дана програмна система є багатофункціональною у використанні, має доступний інтерфейс, що дозволяє працювати з програмою користувачам з різним рівнем комп'ютерної грамотності, має середні вимоги до апаратного та програмного забезпечення, забезпечує досить високу швидкість роботи, безпечна у використанні, що є явними плюсами цього додатка. Отримана система задовольняє заявленим до неї вимогам.

У наступних версіях плануються додати більше можливостей для інтеграції мобільних сенсорів з комп’ютером користувача, наприклад керування яскравістю монітору комп’ютера за допомогою датчика освітленності, розблоковувати ПК за допомогою датчиків телефону. Це дасть більше можливостей розробленій системі і це дозволить зменшити кількість зовнішніх пристроїв, які необхідно купувати для використання комп’ютером.

Отже, це дослідження є актуальним і важливим на даний час. І реалізація даного підходу дозволить кінцевому користувачу зекономити кошти на придбанні додаткових пристроїв та зменшити кількість зовнішніх пристроїв які одночасно під’єднані до комп’ютеру.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. 20 років смартфону: від динозаврів 1990-х - до лідерів 2016 року [Електронний ресурс] – Режим доступу: www/URL: https://www.dw.com/uk/1/g-19470869 – 14.04.2019 – Загл. з екрана

2. W3C DAS WG » Home [Електронний ресурс] – Режим доступу: www/URL: https://www.w3.org/das/ – 14.04.2019 – Загл. з екрана

3. Тренды фронтэнда. Javascript APIs для мобильных устройств / Хабр [Електронний ресурс] – Режим доступу: www/URL: https://habr.com/ru/post/166413/ – 14.04.2019 – Загл. з екрана

4. Exploring the JavaScript Device APIs | Treehouse Blog [Електронний ресурс] – Режим доступу: www/URL: https://blog.teamtreehouse.com/exploring-javascript-device-apis – 14.04.2019 – Загл. з екрана

5. Detecting device orientation - Web APIs | MDN [Електронний ресурс] – Режим доступу: www/URL: https://developer.mozilla.org/en-US/docs/Web/API/Detecting_device_orientation – 14.04.2019 – Загл. з екрана

6. Sun L., Zhang D.: Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. // Ubiquitous Intelligence and Computing. 2010. № 6406. С. 548.

7. Акселерометры Analog Devices – устройство и применение [Електронний ресурс] – Режим доступу: www/URL: https://www.kit-e.ru/articles/sensor/2007_5_46.php – 14.05.2019 – Загл. з екрана

8. Превращаем Android-устройство в пульт ДУ для ПК - ITC.ua [Електронний ресурс] – Режим доступу: www/URL: https://itc.ua/articles/prevrashhaem-android-ustroystvo-v-pult-du-dlya-pk/ – 14.04.2019 – Загл. з екрана

9. Khan A.M., Lee Y.K., Lee S.Y: Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. // International Conference on Future Information Technology (FutureTech). 2010. № 5. С. 6.

10. Davide Anguita, Alessandro Ghio: Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine // International Workshop on Ambient Assisted Living. 2012. № 7658. С. 216.

11. Nishkam Ravi, Nikhil Dandekar: Activity Recognition from Accelerometer Data // Innovative applications of artificial intelligence. 2005. № 17. С. 1541.

12. Перов, А. И. Статистическая теория радиотехнических систем. [Текст] — М.: Радиотехника, 2003. — 400 с.

13. Architecture | WebRTC [Електронний ресурс] – Режим доступу: www/URL: https://webrtc.org/architecture/ – 14.05.2019 – Загл. з екрана

14. Roach, A. B. WebRTC Video Processing and Codec Requirements. — Internet Engineering Task Force, 2016 — 238 с.

15. Perumal M., Wing D., Ravindranath R. — Session Traversal Utilities for NAT (STUN) Usage for Consent Freshness. — Internet Engineering Task Force, 2015 — 184 с.

16. Джозеф Шмуллер. Освой самостоятельно UML 2 за 24 часа. Практическое руководство. [Текст] / Дж. Шмуллер — М.: Вильямс, 2005. — 416 с.

17. Троєлсен Е. C# і платформа .NET. Бібліотека програміста. [Текст]/ Е. Троєлсен — СПб.: Пітер, 2005. — 796 с.

18. Постолит А.В. Visual Studio .NET [Текст] / А.В. Постолит. Разработка приложений баз данных. — СПб.: БХВ-Петербург, 2003. — 544 с.

19. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования. [Текст] / К. Ларман — 3-е изд. — М.: Вильямс, 2006. — 736 с.

20. Daniel Mohl. Building Web, Cloud, and Mobile Solutions With F#. — O'Reilly Media, Inc., 2012 — 105 с.