

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
« ____ » _____ 20__ р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА
ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Мобільний додаток реєстрації спортивних змагань

Освітньо-кваліфікаційний рівень “бакалавр”
Напрямок підготовки 6.050102 – “комп’ютерна інженерія”

Керівник проекту:

(підпис)

Барбарук В.М.,

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Критська Я.О.

(ініціали, прізвище)

Здобувач вищої освіти:

(підпис)

Поддубко С.В.

(ініціали, прізвище)

Група:

КІ-14з

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень бакалавр
Напрямок підготовки 6.050102 – “комп'ютерна інженерія”
(шифр і назва)
Спеціальність _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри КНІ
_____ І.С. Скарга-Бандурова
« _____ » _____ 20 ____ р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Поддубко Сергію Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Мобільний додаток реєстрації спортивних змагань

керівник проекту (роботи) Барбарук Віктор Миколайович, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " _____ " _____ 201_ р. № _____

2. Термін подання студентом роботи 16.06.2018

3. Вихідні дані до роботи дані з проведення змагань з підводного плавання, теоретичні відомості з написання додатків для ОС Android, середа розроблення Eclipse Juno

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз підходів до тестування програмного забезпечення, математична постановка задачі, методи розв'язання та програмна реалізація, охорона праці

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст. викл. Критська Я.О.		

7. Дата видачі завдання 30.04.2018

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз завдання та робота з літературою	05.05.2018 - 13.05.2018	
2	Розробка веб платформи	14.05.2018 - 22.05.2018	
3	Тестування веб платформи	22.05.2018 - 02.06.2018	
4	Розробка розділу «Охорона праці»	02.06 .2018- 11.06.2018	
5	Оформлення пояснювальної записки та електронних плакатів	11.06.2018 - 16.06.2018	

Здобувач вищої освіти

_____ (підпис)

Поддубко С.В.

_____ (прізвище та ініціали)

Керівник

_____ (підпис)

Барбарук В.М.

_____ (прізвище та ініціали)

РЕФЕРАТ

Дипломна робота бакалавра: сторінок 64 , рисунків 9, таблиць 3, джерел 22, додатків 2.

Метою дипломної роботи бакалавра є розробка додатку під Android проведення змагань з плавання та для змоги спостереження бажаними результатами змагань. Актуальність даного мобільного застосування також полягає в тому, що аналогічних додатків немає.

Дипломний проект виконаний з використанням мови програмування Java SDK, в середовищі розробки Eclipse Juno, база даних була побудована у СУБД MySQL.

Ключові слова: модель даних, платформа, додаток, ANDROID.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А,. м. Сєверодонецьк, 93400.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	6
ВСТУП	7
1 ОГЛЯД РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ	10
1.1 Розробка додатків для Android	10
1.1.1 Компоненти додатків в Android	11
1.1.2 Архітектура операційної системи Android.....	14
1.2 Розробка додатків для iPhone	17
1.3 Розробка додатків для Windows Phone	18
1.4 Постановка задачі.....	19
2 РОЗРОБЛЕННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ СПОРТИВНИХ ЗМАГАНЬ З ПІДВОДНОГО ПЛАВАННЯ	20
2.1 Предметна область	20
2.2 Структура Баз даних	21
2.2.1 Засоби та платформи створення баз даних	21
2.2.2 Вибір середовища для розроблення бази даних Swimming	22
2.2.3 Розробка концептуальної моделі БД.....	24
2.3 Використані технології для розробки додатку	25
2.3.1 Мова програмування Java	25
2.3.2 Мова PHP	26
2.3.3 Volley Framework	27
2.3.4 Restfull architecture	28
2.4 Початок роботи з середою розробки Eclipse.....	29
2.4.1 Інструменти Android – розробника	29
2.4.2 Установка Java Development Kit.....	31
2.4.3 Установка Eclipse IDE	32
2.4.4 Установка Android SDK	32
2.4.5 Установка Android Development Tools.....	33

2.4.6 Створення проекту в Eclipse	34
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ДОДАТКУ	37
4 ОХОРОНА ПРАЦІ	41
4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектowanego об'єкту, що мають вплив на персонал	41
4.2 Заходи щодо техніки безпеки	43
4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці.....	46
4.4 Рекомендації по пожежній безпеці.....	50
ВИСНОВКИ.....	54
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	55
ДОДАТОК А. Лістинг коду	57
ДОДАТОК Б. Електронні плакати	60

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

WP - Windows Phone

REST - Representational state transfer

JDK - Java Development Kit

SDK - Software Development Kit

ADT - Android Development Toolkit

IDE - Integrated Development Environment

AOSP - Android Open Source Project

ВСТУП

Головний напрямок вдосконалення мобільних телефонів можна визначити одним поняттям: конвергенція технологій. Мобільні телефони об'єднали в собі чи не все, що можна і розумно поєднувати під корпусом одного пристрою.

Дивлячись на сьогоднішній мобільний телефон, на яку-небудь до межі функціональну і модну модель, можна подумати, що це - межа досконалості, але згадайте, як ми дивилися на перші телефони. Напевно, дивлячись на телефон в 2020-му, ми з посмішкою будемо згадувати "досконалі" апарати 2013-го.

Основні завдання телефону минулого - здійснювати і приймати дзвінки, писати sms. Сьогодні ці завдання доповнилася роботою з інтернет-ресурсами, прослуховуванням музики, фотозйомкою, використанням ігор і додатків. Мобільний майбутнього, швидше за все, додасть до списку основних завдань перегляд телепрограм, управління різною технікою, функції контролю стану здоров'я свого власника і можливо багато чого іншого.

Деякі з перерахованих функцій вже існують в сучасних телефонах, але поки вони відносяться до розряду екзотичних, а прогрес в області їх створення, швидше за все, переведе найбільш потрібні з них в область буденних. Швидше за все, у апаратів майбутнього можуть з'явитися такі можливості, про які ми сьогодні навіть не підозрюємо.

У майбутньому поміняються технології виготовлення телефонів. Зміни технологій можуть бути настільки суттєвими, що, побачивши стільниковий телефон 2016 року, не відразу зрозумієш, що це таке. Причому особлива принадність очікування змін полягає в тому, що, завдяки науково-технічним революціям можуть з'явитися такі пристрої, які зараз не те що здаються неможливими, а просто не розглядаються нами, перебуваючи "за горизонтом" нашої уяви.

Сьогодні на території нашої країни активно експлуатується велика кількість номерів мобільних операторів, про що ще десять років тому вітчизняні провайдери могли тільки мріяти. Ринок мобільних технологій розвивається настільки швидко, що за його змінами встигають не всі, навіть дуже великі компанії. Останні п'ять років показали, що сучасний оператор зв'язку повинен бути максимально гнучким, тримати руку на пульсі світових новинок і винаходів. Той, хто сповідує консервативний підхід до розвитку, досить швидко почне втрачати ринкові позиції, будучи змушеним змінитися або поступитися місцем іншим компаніям.

Експерти звертають увагу споживачів на той факт, що зараз на ринку є кілька мобільних платформ, стрімку популярність серед яких останнім часом набуває iOS і Android. На ринку США компанія Apple вже давно зайняла пальму першості в сегменті дорогих і презентабельних мобільних пристроїв, постійно дивуючи клієнта цікавими новинками. На прикладі Apple, видно як кардинально може змінитися той чи інший ринок з появою нових "девайсів".

Перспективи розвитку ринку мобільних технологій вельми сприятливі, оскільки змінюється не стільки сам ринок, скільки споживач. Якщо вчора мрією багатьох був безлімітний тариф, то сьогодні з широким розповсюдженням мобільного інтернету прерогативи істотно змінилися. З появою на вітчизняному ринку 3G рішень, відбулося помітне перерозподіл клієнтів на користь тих компаній, які були готові до запуску даної послуги. Сучасному власникові мобільного пристрою вже недостатньо просто спілкуватися, йому потрібен цілодобовий доступ в глобальну мережу, за допомогою якої можна не тільки знаходити потрібну інформацію, але і спілкуватися з друзями, повідомляти про своє місцезнаходження та багато іншого.[6]

Що б не говорили, а мобільність сьогодні - явище глобальне, проникаюче в багато сфер нашого життя. Нині споживач шукає не просто мобільний телефон, він уважно вивчає умови, співвідносячи їх з вимогами

сучасного ринку. Саме тому можна стверджувати, що розвиток мобільних технологій буде тільки прискорюватися.

Нижче в дипломній роботі проводиться дослідження в області проведення змагань з підводного плавання на базі мобільних інтерфейсів. Будуть вивчені підходи до реалізації подібних систем. Результатом дипломної роботи стане реалізація програми (Додаток для проведення змагань з підводного плавання) для мобільних пристроїв на платформі Android. У висновку розглядаються перспективи даного дослідження.

1 ОГЛЯД РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ

1.1 Розробка додатків для Android

Додатки під операційну систему Android розробляються в основному з використанням Java. Скомпільований програмний код упаковується у спеціальний файл-архів, Android Package. Цей файл має розширення * .apk і упаковується спеціальною утилітою aapt tool. Саме він надалі поширюється як програма і інсталується на мобільні пристрої. Один такий файл пов'язаний з кодом однієї програми. І кожен додаток в Android живе у своєму власному світі - в такій машині. За замовчуванням, кожна програма виконується в своєму власному процесі, управлінням якого займається ядро Linux, яке також здійснює менеджмент пам'яті. Таким чином, найчастіше код програми виконується в ізоляції від усіх інших додатків. Android стартує процес, коли виникає необхідність виконати який-небудь програмний код і завершує його, коли в ньому більше немає необхідності і системні ресурси потрібні іншим додаткам. За замовчуванням, кожному додатку присвоюється свій унікальний ID Linux-користувача. Права доступу встановлюються таким чином, щоб файли програми були видні тільки цьому користувачеві і даному додатку. Хоча, є способи, що дозволяють експортувати їх в інші програми. Наприклад, існує можливість «поділу» одного і того ж користувальницького ID між двома додатками. У такому випадку, вони зможуть бачити файли один одного. Для того, щоб економити системні ресурси, додатки з однаковим ID можна також домовитися запускати в одному і тому ж Linux-процесі, розділяючи одну і ту ж віртуальну машину.[1]

1.1.1 Компоненти додатків в Android

Цікавою рисою операційної системи Android є той факт, що додаток в ній може використовувати для свого функціоналу елементи інших додатків, якщо, звичайно, вони надають такий доступ. Наприклад, якщо нашому додатку потрібно відобразити прокручуваний список зображень, а інший додаток вже має реалізований відповідний скроллер, відкритий для інших програм, то можна просто викликати його для здійснення цієї дії і не розробляти свій власний. При цьому код нашого застосування не змішується з кодом іншого й не компонується з ним. Швидше, він просто запускає деякий шматок іншого файлу, коли виникає така необхідність. Для забезпечення такого принципу роботи, система повинна бути здатна запускати процес програми тоді, коли буде потрібно будь-яка його частина і створювати екземпляри java-об'єктів саме для цієї частини. Тому додатки Android не мають єдиної точки входу, як це прийнято в більшості систем. Замість цього їх код являє собою набір деяких окремих цілісних сутностей, компонент, з яких система в міру необхідності може створювати екземпляри і використовувати їх.[3]

Можна виділити чотири типи таких сутностей-компонент:

1) Activities. Activity являє собою зовнішній інтерфейс для однієї операції, яку може зробити користувач. Якщо спростити, то це просто один поточний екран як деяка одиниця активності, свого роду кадр з одним користувальницькою дією. Тут і далі буде використовуватися термін activity без перекладу, як деякий ім'я власне. Хоча дослівний переклад, «активність» або «дія», добре передає загальний зміст компонента. Наприклад, activity може надати список пунктів меню, які може вибрати користувач або відобразити фотографії з їхніми підписами. Або інший приклад - додаток для миттєвого обміну повідомленнями може використовувати одне activity для того щоб відобразити лист контактів, інше - щоб створити повідомлення для

обраного контакту, третє - щоб подивитися історію повідомлень або виконати налаштування і так далі.

Всі activity поточного додатка працюють разом і формують єдиний користувальницький інтерфейс, однак при цьому вони незалежні між собою. Кожне з них реалізовано як підклас базового класу Activity, що забезпечує створення вікна, в якому програміст може помістити візуальний інтерфейс.

Додаток може складатися з усього одного activity або відразу з декількох, як згаданий раніше як приклад месенджер. Якими саме будуть activity і скільки їх буде, залежить від конкретного додатка і його дизайну. Як правило, одне з activity позначається як перше, це означає, що воно буде надано користувачеві при запуску програми. Одне activity може запускати інше. Таким чином, перехід від одного activity до іншого здійснюється тоді, коли поточне activity викликає наступне.

Кожне activity надає вікно за умовчанням. Зазвичай вікно створюється в повноекранному вигляді, але воно також може і не займати весь екран і перебувати поверх інших вікон. Activity також може задіяти додаткові вікна - наприклад, спливаюче діалогове для взаємодії з користувачем в процесі роботи activity, або вікно для надання поточної інформації при виборі якої-небудь важливої опції.

Візуальне зміст вікна будується за допомогою ієрархії візуальних компонентів (або подань) - об'єктів, похідних від базового класу View. Кожен компонент являє собою просто прямокутний простір всередині вікна. Батьківські компоненти містять дочірні та організовують їх розташування. Ієрархію компонент можна представити у вигляді дерева, а ті елементи, які знаходяться в самій нижній його частині ("листя") і не мають дочірніх компонент, малює прямокутні області та очікують дій користувача на цій ділянці. Таким чином здійснюється інтерактивна взаємодія з користувачем. Наприклад, таке подання може відображати на екрані маленьку іконку і ініціювати будь-яка дія, коли користувач на неї натисне. В операційній системі Android вже є набір готових візуальних компонент, які доступні для

використання розробниками. Набір включає в себе кнопки, текстові поля, смуги прокрутки, меню, прапорці-перемикачі та багато іншого.

Для того щоб помістити у вікно таку ієрархію, потрібно викликати метод `Activity setContentView ()`. Параметром методу є екземпляр класу `View`, що лежить у корені ієрархії.

2) `Services`. `Services` (послуги) представляють із себе компоненти, які працюють у фоновому режимі. Він, як правило, потрібно для тривалих операцій або для забезпечення роботи віддалених процесів, але в загальному випадку це просто режим, який функціонує, коли додаток не в фокусі. Прикладом такого процесу може стати прослуховування музики в той час, коли користувач робить щось інше або отримання даних по мережі без блокування поточної активності. Сервіс сам по собі не надає користувальницького інтерфейсу, тобто з користувачем не взаємодіє, а запускається, управляється і пов'язаний з іншими компонентами, наприклад, `activity`. Також може запускатися разом з системою.

3) `Content providers`. Даний компонент управляє наборами даних, які додатки надають іншим. Ці дані можуть зберігатися в файлової системі, базах даних `SQLite`, в мережі, або в будь-якому іншому постійному місці, до якого додаток може мати доступ. За допомогою `content provider` інший додаток може запитувати дані і, якщо виставлені відповідні дозволи, змінювати їх. Наприклад, система `Android` містить `content provider`, який управляє користувальницької інформацією про контакти. Він дозволяє будь-якому додатком, що володіє відповідними правами викликати складові цього компонента для того, щоб зчитувати, записувати або змінювати інформацію про конкретну людину.

У більш загальному випадку, `content provider` можна використовувати для читання і запису даних, які використовуються додатком і не є відкритими для інших. Наприклад, додаток `Note Pad` використовує такий компонент для збереження зроблених записів.

Дані компоненти реалізуються як підклас `ContentProvider`. І для того, щоб інші програми могли скоїти операції з даними, їм необхідно надати стандартний набір API.

4) `Broadcast receivers`. Цей компонент відповідає за поширення загальносистемних повідомлень, відстеження та реагування на дії. Багато оповіщення йдуть від системи, наприклад, повідомлення про те що заряд батареї малий або екран вимкнений. Додатки також можуть ініціювати такі оповіщення, наприклад, сигналізувати про те, що інформація завантажена на пристрій і доступна до використання. Як і сервіси, `broadcast receiver` не надає користувальницького інтерфейсу, проте, він здатний створювати повідомлення в рядку стану, щоб попереджати користувача про те, що сталося якусь подію. Проте частіше `broadcast receiver` взаємодіє з іншими компонентами для того, щоб самому виконувати мінімальний обсяг роботи. Так, він може ініціювати сервіси для виконання дій, прив'язаних до якоїсь події.

1.1.2 Архітектура операційної системи Android

Розглянемо архітектуру операційної системи Android (Рис. 1.1).

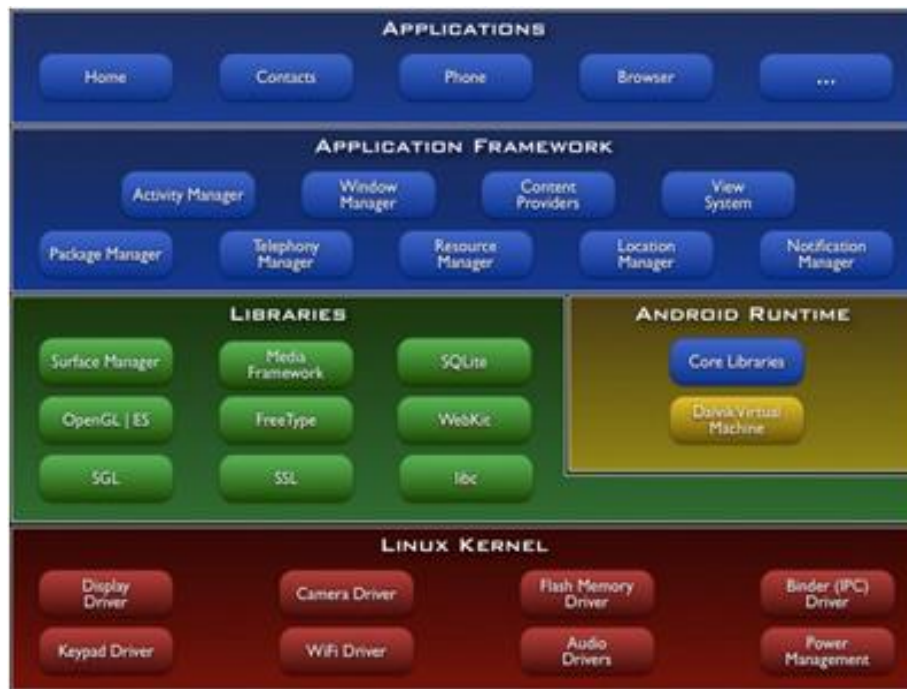


Рисунок 1.1 – Архітектура операційної системи Android

Якщо уявити компонентну модель Android у вигляді деякої ієрархії, то в самому низу, як сама фундаментальна і базова складова, розташовуватиметься ядро операційної системи.

Часто компонентну модель ще називають програмним стеком. Дійсно, це визначення тут доречно, тому що йдеться про набір програмних продуктів, які працюють разом для отримання підсумкового результату. Дії в цій моделі виконуються послідовно, і рівні ієрархії також послідовно взаємодіють між собою.

Як відомо, Андроїд заснований на кілька урізаному ядрі ОС Linux і тому на цьому рівні ми можемо бачити саме його (версії 2.6.x). Воно забезпечує функціонування системи і відповідає за безпеку, управління пам'яттю, енергосистемою і процесами, а також надає мережевий стек і модель драйверів. Ядро також діє як рівень абстракції між апаратним забезпеченням і програмним стеком.

«Вище» ядра, як програмне забезпечення проміжного шару, лежить набір бібліотек (Libraries), призначений для забезпечення найважливішого базового функціоналу для додатків. Тобто саме цей рівень відповідає за

надання реалізованих алгоритмів для верхніх рівнів, підтримку файлових форматів, здійснення кодування і декодування інформації (в приклад можна привести мультимедійні кодеки), отрисовку графіки та багато іншого. Бібліотеки реалізовані на C / C ++ і скомпільовані під конкретне апаратне забезпечення пристрою, разом з яким вони і поставляються виробником в передвстановленому вигляді.

На цьому ж рівні розташовується Android Runtime - середовище виконання. Ключовими її складовими є набір бібліотек ядра і віртуальна машина Dalvik. Бібліотеки забезпечують більшу частину низкоуровневої функціональності, доступної бібліотекам ядра мови Java.

Кожна програма в ОС Android запускається у власному екземплярі віртуальної машини Dalvik. Таким чином, всі працюючі процеси ізольовані від операційної системи і один від одного. І взагалі, архітектура Android Runtime така, що робота програм здійснюється строго в рамках оточення віртуальної машини. Завдяки цьому здійснюється захист ядра операційної системи від можливої шкоди з боку інших її складових. Тому код з помилками або шкідливе ПЗ не зможуть зіпсувати Android і пристрій на його базі, коли спрацюють. Така захисна функція, поряд з виконанням програмного коду, є однією з ключових для надбудови Android Runtime. Рівнем вище розташовується Application Framework, іноді званий рівнем каркаса додатків. Саме через каркаси додатків розробники отримують доступ до API, що надаються компонентами системи, що лежать нижче рівнем. Крім того, завдяки архітектурі фреймворку, будь-якого додатка надаються вже реалізовані можливості інших програм, до яких дозволено отримувати доступ.

Таким чином, завдяки Application Framework, додатки в ОС Android можуть отримувати в своє розпорядження допоміжний функціонал, завдяки чому реалізується принцип багаторазового використання компонентів додатків і операційної системи. Природно, в рамках політики безпеки.

Варто відзначити, просто на понятійному рівні, що фреймворк лише виконує код, написаний для нього, на відміну від бібліотек, які виконуються самі. Ще одна відмінність полягає в тому, що фреймворк містить в собі велику кількість бібліотек з різною функціональністю і призначенням, в той час як бібліотеки об'єднують в собі набори функцій, близьких за логікою.

На вершині програмного стека Android лежить рівень додатків (Applications). Сюди відноситься набір базових додатків, який предумовлен на ОС Android. Наприклад, в нього входять браузер, поштовий клієнт, програма для відправки SMS, карти, календар, менеджер контактів і багато інших. Список інтегрованих програм може змінюватися в залежності від моделі пристрою та версії Android. І крім цього базового набору до рівня додатків відносяться в принципі всі програми під платформу Android, в тому числі і встановлені користувачем. Вважається, що програми під Android пишуться на мові Java, але потрібно зазначити, що існує можливість розробляти програми і на C / C ++ (за допомогою Native Development Kit), і на Basic (за допомогою Simple) і з використанням інших мов. Також можна створювати власні програми за допомогою конструкторів додатків, таких як App Inventor. Словом, можливостей тут багато. Ну і ще раз про архітектуру ОС Android можна дізнатися з перших рук, подивившись це відео, в якому розробники представляють презентацію, присвячену розбору системи.

1.2 Розробка додатків для iPhone

iOS - це оптимізований під мобільні пристрої варіант Mac OS X, розроблений Apple Inc. Він встановлюється на всі пристрої iPod touch, iPhone і iPad, і займає всього біля 500 мб дискового простору.

Swift - мультіпарадигменна об'єктно-орієнтована мова програмування, створена компанією Apple для розробників iOS і OS X. Swift працює з

фреймворками Cocoa і Cocoa Touch і сумісний з основною кодовою базою Apple, написаної на Objective-C. Swift замислювався як більш безпечна мова в порівнянні з Objective-C. Мова підтримується в середовищі програмування Xcode 6 програми на ньому компілюються за допомогою Apple LLVM і використовують рантайм Objective-C, що робить можливим використання обох мов (а також pure C і C++) в рамках однієї програми. Був анонсований на конференції розробників WWDC 2014.

1.3 Розробка додатків для Windows Phone

Всі програми для Windows Phone створюються з використанням керованого коду .NET. В даний час C# - єдина підтримувана мова програмування. Вільно доступний для завантаження Microsoft Visual Studio 2010 Express для Windows Phone включає XNA Game Studio 4.0 і екранний емулятор телефону, а також інтегрується з Visual Studio 2010.

Візуальні елементи та анімація для додатків Silverlight можуть створюватися у Microsoft Expression Blend. Платформи Silverlight і XNA для Windows Phone мають ряд загальних бібліотек, тобто деякі бібліотеки XNA можуть використовуватися в програмі Silverlight і навпаки. Але не можна створювати програму, що поєднує в собі візуальні елементи обох платформ.

Як правило, Silverlight використовується для програм, які можна класифікувати як додатки або утиліти. Опис компоновки елементів управління і панелей для користувача інтерфейсу в цих програмах виконується за допомогою розширювана мова розмітки додатків (Extensible Application Markup Language, XAML). У файлах виділеного коду можуть реалізовуватися операції з ініціалізації і деяка логіка, але основним їх призначенням є обробка подій елементів управління. Silverlight дозволяє реалізовувати в Windows Phone стиль насичених інтернет-додатків (Rich

Internet Applications, RIA), включаючи мультимедіа та Веб. Для Windows Phone створена версія Silverlight 3, в яку не ввійшли деякі можливості, що не підходять для телефону, але компенсовані поруч доповнень.

Головне призначення XNA - створення високопродуктивних ігор. Для 2D-ігор спрайт і підкладки описуються за допомогою растрових зображень; для 3D ігор створюються тривимірні моделі. Дія гри, що включає переміщення графічних об'єктів по екрану і запит користувача введення, обробляється вбудованим ігровим циклом XNA.

1.4 Постановка задачі

Необхідно створити додаток під Android, за допомогою якого можна буде проводити змагання з підводного плавання. Цей додаток має дозволяти заходити в систему управління під своїм логіном та обрати змагання, інформацію про які ви хочете подивитися, після чого повинен з'являтися список днів змагань та їх видів. За допомогою програми можна спостерігати за стартовим протоколом та результуючим, обираючи номер запливу. Для випадку, коли інтернет недоступний, необхідно створити внутрішню базу даних.

2 РОЗРОБЛЕННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ СПОРТИВНИХ ЗМАГАНЬ З ПІДВОДНОГО ПЛАВАННЯ

2.1 Предметна область

Швидкісне плавання - окремий спорт, де спортсмени змагаються на різних дистанціях у плаванні за допомогою моноласту або класичних ласт. Деякі дистанції проходять під водою з кисневим балоном в руках, деякі - на поверхні води з використанням трубки. У 1986 році швидкісне плавання в ластах було офіційно визнано Міжнародним Олімпійським комітетом, що послужило двигуном до популяризації цього виду спорту в світі. Швидкісне плавання в моноласті з'явилося після того, як вперше в 70-х роках минулого століття в СРСР зробили моноласту з титану і гумових ласт. Швидкісне плавання в моноласті дає можливість досягати швидкості у воді до 8 км /год.

Підводне плавання включає такі види змагань:

- 50 м пірнання;
- 50 м плавання в ластах;
- 100 м плавання в ластах;
- 200 м плавання в ластах;
- 400 м плавання в ластах;
- 800 м плавання в ластах;
- 1500 м плавання в ластах;
- 50 м плавання в біластах;
- 100 м плавання в біластах;
- 200 м плавання в біластах;
- 400 м плавання в біластах;
- 100 м підводне плавання;
- 400 м підводне плавання;
- 800 м підводне плавання;

- естафета 4x100 м;
- естафета 4 x200 м;
- естафета 4 x50 м;
- марафон 3000 м;
- 25 м плавання в біластах;
- 25 м плавання в ластах;
- 25 м пірнання.

Основне завдання плавця у спортивному плаванні - підготуватися до швидкісного подолання дистанції і показати свій максимально можливий результат на змаганнях.

2.2 Структура Бази даних

2.2.1 Засоби та платформи створення баз даних

У наш час існує велика кількість програмного забезпечення, котре дозволяє створювати, редагувати та працювати з базами даних. Найпоширенішими є : MySQL, MDB viewer Plus, Borland Database Engine, phpMyAdmin, Firebird та інші. Вони працюють з різним програмним забезпеченням баз даних, такими, як Oracle, MySQL або SQL server.

Oracle (Oracle Corporation) - американська корпорація, найбільший в світі розробник програмного забезпечення для організацій, великий постачальник серверного обладнання. Oracle Corporation (nasdaq ORCL) - одна з найбільших американських компаній, розробників систем керування базами даних, інструментів для розробки баз даних, а також ERP-систем.

MySQL - вільна система керування реляційними базами даних. MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з

самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL - одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

Microsoft SQL Server - комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів - Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI/ISO щодо структурованої мови запитів (SQL) із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.

2.2.2 Вибір середовища для розроблення бази даних Swimming

База даних Swim для розробленого додатку створена на платформі MySQL за допомогою програмного забезпечення Denwer.

Denwer - програмне середовище, призначена для розробки web-додатків на базі популярної мови програмування PHP, web-сервера Apache і бази даних MySQL на локальному сервері. Даний пакет, також іменованій джентльменським набором web розробника, був створений в 1999 році групою розробників під керівництвом Дмитра Котеровим як комплекс додатків для web-розробки з автоматизованою установкою і налаштуванням вище зазначених компонентів. Остання на сьогодні версія - Denwer 3. Вона включає в себе PHP версії 5.3, Apache 2.2, MySQL 5.5, phpMyAdmin для управління базами даних, системи управління запуском і віртуальними хостами, емулятор пошти і установник. Також даний комплекс може бути

розширений додатковими модулями PHP і SQL, а також мовами Perl і Python, базами даних MS SQL, Postgre SQL та іншими компонентами.

Переваги даної платформи:

- швидкодія. СУБД MySQL є однією з найшвидших баз даних з наявних на сучасному ринку;

- простота використання. СУБД MySQL є високопродуктивною і відносно простою у використанні СУБД, яку значно простіше інсталювати і адмініструвати, ніж багато великих систем;

- підтримка мови запитів. MySQL "розуміє" команди мови SQL, мови запитів, яка знайшла застосування у всіх сучасних СУБД;

- можливості обробки. Кількість рядків у таблицях може досягати 50 млн. Компанія-розробник стверджує, що використовує MySQL з 1996 року на сервері з більш ніж 40 базами даних, які містять 10000 таблиць, з яких більш ніж 500 мають більш 7 мільйонів рядків;

- можливості доступу. Сервер дозволяє одночасно підключатися необмеженій кількості користувачів, що одночасно працюють з базою даних. Доступ до сервера СУБД MySQL можна здійснити в інтерактивному режимі;

- доступ до баз даних СУБД MySQL можливий і за допомогою додатків, що підтримують ODBC (Open Data Bases Connectivity, відкрите взаємодію з базами даних) - узагальнений інтерфейс, який може використовуватися для одночасного з'єднання з різними СУБД, розробленими компанією Microsoft;

- взаємодія і безпека. MySQL призначена для роботи в мережі і може бути доступна через Internet. Таким чином, до сервера MySQL можна підключатися з будь-якої точки земної кулі через програму-клієнт, встановлену на комп'ютері користувача. Але при цьому СУБД MySQL забезпечена розвиненою системою захисту від несанкціонованого доступу. Для забезпечення додаткового захисту СУБД MySQL підтримує кріптіруєме з'єднання з використанням протоколу SSL;

– - апаратна сумісність. СУБД MySQL відмінно працює як під управлінням самих різних версій UNIX, так і під управлінням систем, таких як Windows і OS / 2. СУБД MySQL може працювати як на домашніх ПК, так і на потужних серверах;

– - малий розмір. СУБД MySQL має обмежений розмір, особливо в порівнянні з величезним дисковим простором, необхідним більшості комерційних СУБД (Сервер Apache - 23 Мбайт, СУБД MySQL - 120 Мбайт, PHP - 150 Мбайт).

2.2.3 Розробка концептуальної моделі БД

Концептуальний рівень - узагальнююче уявлення бази даних. Цей рівень описує те, які дані зберігаються в базі даних, а також зв'язки, що існують між ними.

Даний рівень є проміжним у трирівневої архітектурі. Він включає логічну структуру всієї бази даних

На концептуальному рівні представлені такі компоненти:

- - всі сутності, їх атрибути та зв'язку;
- - накладаються на дані обмеження;
- - семантична інформація про дані;
- - інформація про заходи забезпечення безпеки і підтримки цілісності даних.

Ґрунтуючись на описі предметної області та постановки завдання курсового проекту, можна виділити наступні суті, описані в таблиці 2.1

Таблиця 2.1 - Опис сутностей

Назва сутностей	Первинний ключ
Змагання	ІН_Змагань
Вид_Змагання	ІН_Вид_Змагань
Дистанції	ІН_Дістанції
Учасники	ІН_Учасника
Тренер	ІН_Тренера
Команда	ІН_Команди
Заплив	ІН_Запливу

На підставі таблиці сутностей і постановки завдання дипломного проекту, виділяються всі необхідні атрибути.

2.3 Використані технології для розробки додатку

2.3.1 Мова програмування Java

Java - об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java. Зараз мовою займається компанія Oracle, яка придбала Sun Microsystems у 2009 році. Синтаксис мови багато в чому схожий на C та C++. У офіційній реалізації, Java програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Oracle надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.[2]

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки

програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім, Java розроблялась як платформи-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням. За необхідності таких дій java дозволяє викликати підпрограми, написані іншими мовами програмування.

Java вплинула на розвиток J++, що розроблялась компанією Microsoft. Роботу над J++ було зупинено через судовий позов компанії Sun Microsystems, оскільки ця мова програмування була модифікацією Java. Пізніше в новій платформі Microsoft .NET випустило J#, щоб полегшити міграцію програмістів J++ або Java на нову платформу. З часом нова мова програмування C#, стала основною мовою платформи, перейнявши багато чого з Java. J# востаннє включався в версію Microsoft Visual Studio 2005. Мова сценаріїв JavaScript має схожу із Java назву і синтаксис, але не пов'язана із Java.

2.3.2 Мова PHP

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

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

використовувати PHP для генерування і JavaScript-кодів які виконуються вже на стороні клієнта.

2.3.3 Volley Framework

Volley це HTTP бібліотека, яка робить мережевий обмін Android додатків більш легким і, найголовніше, більш швидким. Volley доступний для використання через відкрите AOSP сховище.

Volley пропонує наступні переваги:

- автоматичне планування мережевих запитів;
- кілька паралельних мережевих підключень;
- прозоре кешування на диску і в пам'яті відповідей, використовуючи стандартну HTTP когерентність кеша;
- підтримка пріоритетів запитів;
- API для скасування запиту. Ви можете скасувати один запит, або ви можете встановити блокування або область запитів для скасування;
- простота настройки, наприклад, для повторних спроб і відстрочки;
- строгий порядок, який дозволяє легко і правильно заповнити користувацький інтерфейс даними отриманих з мережі асинхронно;
- інструменти для налагодження та трасування;

Volley виділяється серед RPC операцій, використовуваних для заповнення користувача інтерфейсу, таких як отримання сторінки результатів пошуку у вигляді структурованих даних. Вона легко інтегрується з будь-яким протоколами і поставляється з підтримкою сирих рядків, зображень та JSON. Надаючи вбудовану підтримку функції, які вам необхідні, Volley звільняє вас від написання стандартного коду і дозволяє зосередитися на логіці, яка специфічна для вашої програми.[4]

2.3.4 Restfull architecture

REST - це стиль архітектури програмного забезпечення для розподілених систем, таких як World Wide Web, який, як правило, використовується для побудови веб-служб. Термін REST був введений в 2000 Роєм Філдінгом, одним з авторів HTTP-протоколу. Системи, що підтримують REST, називаються RESTful-системами.

У загальному випадку REST є дуже простим інтерфейсом управління інформацією без використання якихось додаткових внутрішніх прошарків. Кожна одиниця інформації однозначно визначається глобальним ідентифікатором, таким як URL. Кожна URL в свою чергу має строго заданий формат.

REST використовується для побудови Web-сервісів. Для кожної одиниці інформації (info) визначається 5 дій. А саме:

- GET / info / (Index) - отримує список всіх об'єктів. Як правило, це спрощений список, тобто містить лише поля ідентифікатора і назви об'єкта без інших даних;
- GET / info / {id} (View) - отримує повну інформацію про об'єкт;
- PUT / info / або POST / info / (Create) - створює новий об'єкт. Дані передаються в тілі запиту без застосування кодування;
- POST / info / {id} або PUT / info / {id} (Edit) - змінює дані з ідентифікатором {id}, можливо замінює їх;
- DELETE / info / {id} (Delete) - видаляє дані з ідентифікатором {id}.

Як видно, архітектура REST дуже проста у плані використання. По виду запиту, що надійшов відразу можна визначити, що він робить, не розбираючись у форматах (на відміну від SOAP, XML-RPC). Дані передаються без застосування додаткових шарів, тому REST вважається менш ресурсоємним.

2.4 Початок роботи з середою розробки Eclipse

2.4.1 Інструменти Android – розробника

Безпосередньо перехід до практики програмування під будь-яку платформу починається з розгляду наявного арсеналу інструментів розробки. Вже після того, як інструменти обрані, налаштовані для роботи і випробувані в написанні яких-небудь простих інструкцій, починається програмування та вдосконалення навичок.[5]

Перш ніж розглядати інструменти для Android, необхідно ввести деякі загальні поняття, з якими належить зіткнутися на даному етапі. Перше базове поняття - це SDK або Software Development Kit («devkit»). Під ним розуміють набір засобів розробки, що дозволяє програмістам створювати додатки під деяку архітектуру - для певного пакету програм, апаратної або програмної платформи, операційної або комп'ютерної систем. SDK, як правило, надає розробнику широкий набір інструментів, який може включати в себе засоби налагодження та довідкові матеріали (приклади коду, замітки і т.д.). SDK часто надається безпосередньо розробником цільової платформи і в ньому враховуються її особливості. Поширюватися (як і у випадку з Android) такий інструментарій може безкоштовно. Це допомагає платформі обзаводитися різноманітними додатками за рахунок сторонніх розробників.

Ще одним поняттям, яке часто можна зустріти, коли йдеться про інструменти розробки, є IDE (Integrated development environment) - інтегроване середовище розробки. Вона являє собою набір програмних засобів, покликаний максимізувати продуктивність програміста за рахунок побудови для нього зручною і доброзичливою середовища для процесу програмування. Цей складний програмний комплекс може включати в себе текстовий редактор (найчастіше з підсвічуванням синтаксису і підказками), компілятор або інтерпретатор (іноді підтримуються кілька мов), інтегрований відладчик і засоби для автоматизації збирання. Часто підтримуються

можливості інтеграції зі стороннім програмним забезпеченням, наприклад, засобами проектування або контролю версій. Також найчастіше в IDE є інструменти для швидкої побудови і візуального редагування графічного інтерфейсу майбутньої програми, а також інструменти на зразок діаграм ієрархії класів, браузера класів, інспектора об'єктів або менеджера ресурсів. Таким чином, інтегрована середовище розробки покликана об'єднати різні інструменти в один програмний комплекс для забезпечення зручності та підвищення швидкості розробки ПЗ. Хоча на ділі можна і не використовувати для програмування яку-небудь IDE, віддавши перевагу незв'язні між собою інструменти.

Розробка додатків для платформи Android пов'язана з групою інструментів, які надаються набором Android SDK. Також знадобляться інструментарій для розробки додатків в Java SE (JDK) і інтегрована середовище розробки. В якості останньої прийнято використовувати Eclipse IDE, що будемо робити і ми, хоча варто відзначити, що розробку додатків можна вести і за допомогою найпростішого текстового редактора або інших IDE, а також викликати інструменти за допомогою скриптів або використання командного рядка. Проте розробка в Eclipse є кращим методом, тому що, по-перше, це середовище може безпосередньо звертатися до необхідних інструментів, а по-друге, для неї існує спеціальний плагін, Android Development Toolkit (ADT), який забезпечує перехід до створення додатків під Android за допомогою Eclipse IDE ще більш зручним чином.

У висновку відзначимо, що Android SDK вимагає JDK версії 5 або вище, а також Eclipse версії 3.3 або вище. На сайтах розробників доступні версії SDK, Java і Eclipse для операційних систем Windows, Linux і Mac OS, а в SDK входить емулятор для кожної з них. Взагалі кажучи, програми під Android виконуються у віртуальній машині, і вибір будь-який з існуючих операційних систем не дає ніяких переваг перед іншими, так що можна працювати з тією, яка більш зручна.

2.4.2 Установка Java Development Kit

Для створення програм на мові Java необхідно спеціальне програмне забезпечення. Самі останні версії цього ПО можна завантажити з офіційного сайту розробника, Oracle Corporation.

До цього програмному комплексу відносяться такі інструменти як JRE (Java Runtime Environment) і JDK (Java Development Kit). Перший інструмент являє собою середовище виконання - мінімальну реалізацію віртуальної машини, в якій запускається і виконується програмний код на Java. Другий інструмент - це в свою чергу цілий набір інструментів, комплект розробника додатків мовою Java. Насправді, JRE також входить до складу JDK, так само як і різні стандартні бібліотеки класів Java, компілятор javac, документація, приклади коду і різноманітні службові утиліти. Весь цей набір поширюється вільно і має версії для різних ОС, тому будь-хто може його завантажити і використовувати.

В JDK не входить інтегроване середовище розробки, передбачається, що її розробник буде встановлювати окремо. Існують численні IDE для Java-розробки, наприклад, NetBeans, IntelliJ IDEA, Borland JBuilder та інші. Але ми обумовили раніше, що для розробки додатків під Android ми виберемо Eclipse IDE. Для установки JDK, спершу необхідно завантажити її з сайту розробника, вибравши потрібну версію для своєї операційної системи і архітектури. Потім проводиться інсталяція даного програмного забезпечення з параметрами за замовчуванням, які пропонуються майстром установки.

2.4.3 Установка Eclipse IDE

Щоб отримати робочий інструментарій для розробки додатків під Android - це завантажити та встановити середу Eclipse. Завантажити дану IDE можна з порталу Eclipse Foundation, з цієї сторінки. Як і інші необхідні нам інструменти, цей програмний продукт вільно поширюється і його можна використовувати на будь-якій операційній системі.

А поки завантажити потрібні нам версію і встановимо її. Досить буде версії «for Java developers», але нічого страшного, якщо це буде більш наворочена «for EE developers». Розпакуємо вміст завантаженого архіву в будь-який зручний місце на жорсткому диску. Наприклад, на Windows можна розпакувати прямо в корінь системного диска (як це зробив я), на Linux - в каталог Home, на Mac OS - в Applications. Якщо б ОС не була обрана, в каталозі Eclipse можна знайти однойменний виконуваний файл (для Windows це eclipse.exe), запустивши який, ми запустимо дану IDE. Зручно винести ярлик для цього файлу на видне місце.

При першому запуску IDE попросить вказати робочий простір (workspace), це одна з концепцій Eclipse. Робочий простір - це контейнер для проектів, що має свій власний набір метаданих і установок (preferences). Наприклад, у мене це буде C: / Android. Після того як середовище розробки завантажиться, на головному екрані потрібно натиснути значок Workbench - Go to the workbench.

2.4.4 Установка Android SDK

Для розробки додатків для платформи Android нам знадобиться встановити на комп'ютер Android SDK. Крім інших інструментів розробника

в нього входить емулятор нашої мобільної операційної системи, так що для програмування і налагодження навіть необов'язково мати під рукою пристрій на базі Android. SDK є у вільному доступі, і завантажити його можна з офіційного сайту - <http://developer.android.com/sdk/index.html>. Звідси потрібно завантажити версію SDK для своєї операційної системи.

Запустимо виконуючий файл з назвою SDK Manager. Тут може виникнути проблема з його роботою. Якщо видаються помилки і додаток не запускається, то тут справа у змінній середовища PATH, яка зберігає в собі шляхи пошуку для виконуваних файлів і використовується різними програмами і скриптами. Для коректної роботи в цій змінній повинен бути вказаний шлях до каталогу інструментів Android SDK, а також до каталогу bin для JDK.

Після запуску програма з'єднається з репозиторієм і у вікні менеджера відобразиться список доступних пакетів. У віконці Available Packages (доступні пакети) виберемо потрібні пакети, після чого потрібно натиснути Install Selected (встановити вбрання). Після успішної установки ми будемо мати на комп'ютері повноцінний бойовий Android SDK і можемо перейти до встановлення та налаштування плагіна для Eclipse.

2.4.5 Установка Android Development Tools

Плагін для Eclipse, Android Development Tools (ADT), допоможе писати програми для Android. Він інтегрується з Eclipse IDE для того, щоб забезпечити можливості створення, налагодження і тестування android-додатків.

Хід установки по кроках:

1. Запускаємо середу Eclipse.

2. У меню вибираємо елемент Help (Допомога), потім пункт Install New Software (Встановити нову програму).

3. У вікні Install, в поле Work with (Працювати з) вводимо адресу ресурсу - <https://dl-ssl.google.com/android/eclipse> і тиснемо Enter. Якщо все зроблено правильно, то Eclipse з'єднається з ресурсом і завантажить звідти список, який відобразиться трохи нижче як запис Developer Tools з декількома підлеглими вузлами.

4. Виберемо (відзначимо галочкою) вузол Developer Tools (також переконаємося, що всі підлеглі вузли також відмічені) і потім натиснемо Next (Далі).

5. З'явиться вікно, в якому потрібно підтвердити установку обраних пакетів. Тиснемо Next (Далі). У наступному вікні нам пропонується прочитати ліцензійну угоду по ADT і пов'язаним з ним інструментам. Щоб продовжити установку, необхідно його прийняти - вибираємо пункт «I assent» (Я згоден) і тиснемо Next (Далі).

Тепер Eclipse автоматично завантажить і встановить плагін ADT, а потім запропонує перезапустити IDE. Це необхідно, щоб плагін відобразився в Eclipse.

2.4.6 Створення проекту в Eclipse

Щоб почати написання коду програми виконаємо наступні дії:

- 1) Запускаємо Eclipse.
- 2) Обираємо елемент меню File, пункт New і потім Project.
- 3) Обираємо Android, розгортаємо його і обираємо підпункт Android Project. Натискаємо Next.
- 4) З'явиться вікно, у котрому потрібно заповнити поля (Рис 2.1).

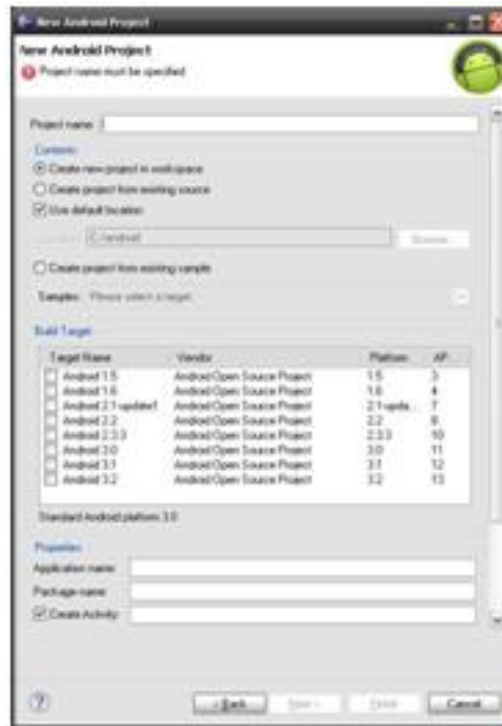


Рисунок 2.1 – Створення проекту

Поле Project Name призначене для імені нашого проекту, впишемо сюди «Swimming». У графі Build Target потрібно вказати, яка версія API буде використана в проекті. Принцип простий, потрібно просто знати, яку функціональність ми будемо використовувати в нашій програмі, і ставити мінімальну версію API, яка дозволить цю функціональність реалізувати. У нашому випадку виберемо версію Android 4.3. У полі Application Name вказується назва виконуючого файлу програми. Залишимо «Swimming», надалі можна буде перейменувати. Package Name означає назва пакету і сюди можна записати унікальну назву нашого проекту серед усіх програм для платформи Android. Хорошим тоном вважається заповнення цього поля у нижньому регістрі, а унікальності можна досягти, дописавши сюди ідентифікатор розробника. Близько графі Create Activity повинна стояти галочка, а в полі потрібно написати назву для цього об'єкта. Залишимо як є («SwimmingActivity»). У Min SDK Version (мінімальна версія SDK) вказується для якої версії платформи створюється додаток, тобто в яких пристроях воно буде підтримуватися. Залишимо 3. Тиснемо кнопку Finish.

5. Ми створили ядро для нашого проекту. Зліва ми можемо бачити вікно Package Explorer. Тут відображаються всі компоненти нашого застосування.

6. Приступаємо до написання коду застосунку.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ДОДАТКУ

На етапі дослідної експлуатації перевіримо придатність системи для кінцевих користувачів. Визначимо основну зручність роботи з системою. Після успішного завершення етапу дослідної експлуатації та використання системи різними користувачами система може класифікуватися як комерційна.

При запуску додатку користувач потрапляє на головну сторінку системи, що зображена на рисунку 3.1. Для того щоб переглянути інформацію про змагання і їх результати, користувач повинен ввести свій логін і пароль, та натиснути кнопку "Войти".

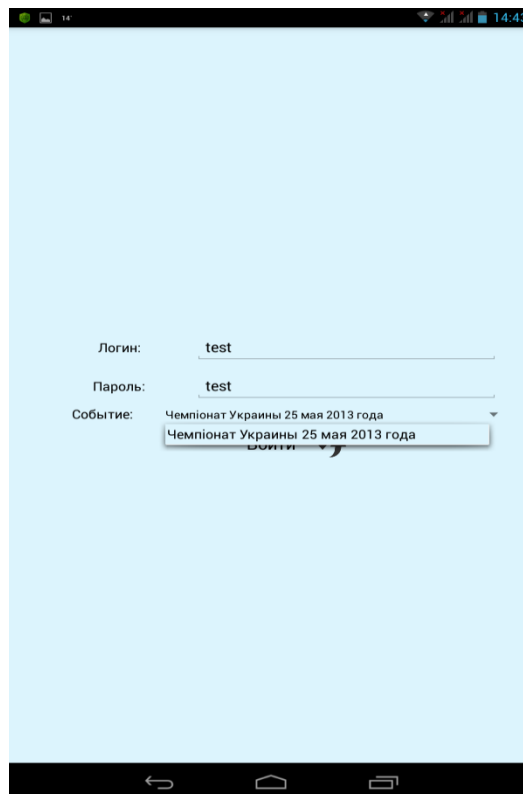


Рисунок 3.1 – Головна сторінка

Якщо дані були введені не вірно, то система виведе повідомлення про неправильне заповнення або логіну, або паролю (Рис.3.2).

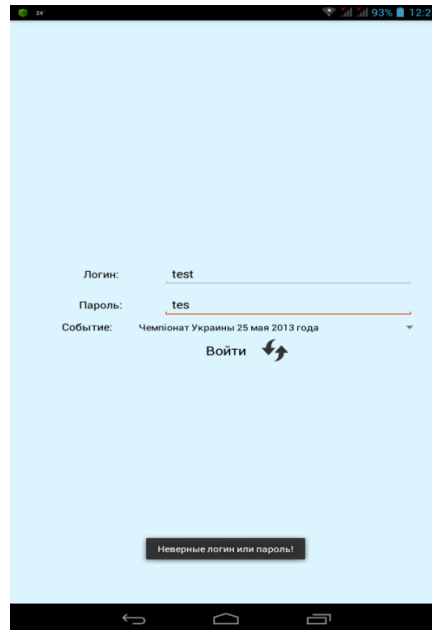


Рисунок 3.2 – Повідомлення про невірно введені дані

Після того, як змагання були вибрані, користувач потрапляє у вікно з меню днів змагань та їх видів (Рис. 3.3).

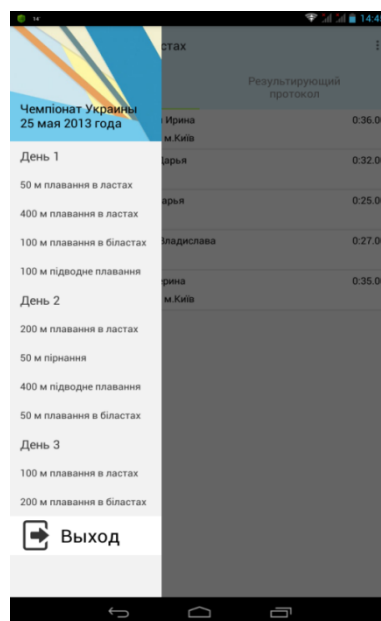


Рисунок 3.3 – Вікно зі списком днів та видів змагань

Після того, як вибраний день та вид змагань, користувач потрапляє на сторінку зі стартовим протоколом (Рис. 3.4).

50 м плавання в ластах		Стартовий протокол	Результуючий протокол
2	Юн. Карбовская Ирина ДЮСШ Аквалідер - ЦСТК м.Київ		0:36.00
3	Юн. Черватюк Дарья Дарниця-303 м.Київ		0:32.00
4	Зр. Савченко Дарья Дарниця-303 м.Київ		0:25.00
5	Зр. Игнатенко Владислава Дарниця-303 м.Київ		0:27.00
6	Юн. Кищак Катерина ДЮСШ Аквалідер - ЦСТК м.Київ		0:35.00

Рисунок 3.4 – Вікно зі стартовим протоколом «50 м плавання в ластах»

50 м плавання в ластах		Стартовий протокол	Результуючий протокол
2	Юн. Карбовская Ирина ДЮСШ Аквалідер - ЦСТК м.Київ		-
3	Юн. Черватюк Дарья Дарниця-303 м.Київ		-
4	Зр. Савченко Дарья Дарниця-303 м.Київ		-
5	Зр. Игнатенко Владислава Дарниця-303 м.Київ		-
6	Юн. Кищак Катерина ДЮСШ Аквалідер - ЦСТК м.Київ		-

Рисунок 3.5 – Вікно з результуючим протоколом «50 м плавання в ластах»

Також користувач може вибрати номер запливу, натиснувши на меню опцій для даного запливу (Рис. 3.6)

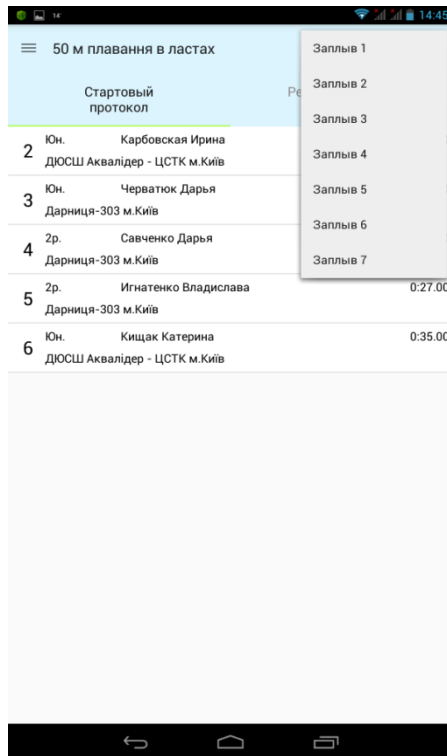


Рисунок 3.6 – Вікно вибору номеру запливу

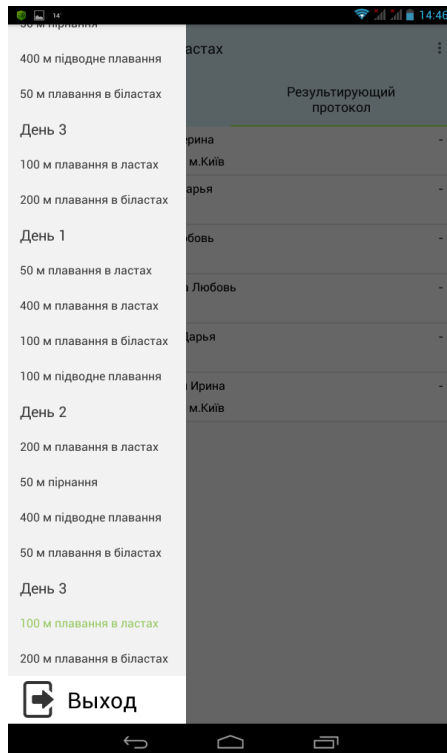


Рисунок 3.7 – Вікно виходу з вибраних замахань

4 ОХОРОНА ПРАЦІ

4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкту, що мають вплив на персонал

У даному дипломному проєкті розробляється програмне забезпечення.

Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Виходячи з приведених характеристик, вочевидь, що для людини існує небезпека поразки електричним струмом, унаслідок недбалого поводження з комп'ютером і порушення правил експлуатації, залишення частин ПЕОМ, що знаходяться під напругою, відкритими або знятих для ремонту вузлів.

Відповідно до [7] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною(невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;

- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу слід приділяти біологічним ефектам низькочастотних електромагнітних полів(аж до порушення ДНК).

Відповідно до [8], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання якої може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищений або знижений рух повітря;
- підвищена або знижена вологість повітря;
- відсутність або недостатність природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

4.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної(нагрів тканин і біологічних середовищ), електролітичної(розкладання крові і плазми) і біологічної(роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом [9]:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне розділення ятерів;
- використання малої напруги;
- ізоляція частин, що проводять струм;
- огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує година, протягом якого людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_k = \frac{U_\phi}{Z_\Pi + \frac{Z_T}{3}}, \quad (4.1)$$

де U_ϕ - номінальна фазна напруга мережі, В;

Z_Π - повний опір петлі, створене фазними і нульовими дротами, Ом;

Z_T - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [10]: $Z_T/3 = 0,1$ Ом.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу(4.2.) :

$$Z_\Pi = \sqrt{R_\Pi^2 + X_\Pi^2}, \quad (4.2)$$

де $R_\Pi = R_\phi + R_0$ - сумарний активний опір фазного R_ϕ і нульового R_0 дротів, Ом;

X_Π - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту $S = 2.5$ мм, тоді згідно таблицям 5 і 6 [18], має такий опір:

$$X_\Pi = 0,11 \text{ Ом};$$

$$R_\phi = 7,55 \text{ Ом};$$

$$R_0 = 7,55 \text{ Ом}.$$

$$\text{Отже, } R_\Pi = 7,55 + 7,55 = 15,1 \text{ Ом}.$$

Тоді по формулі (4.2) знаходимо повний опір петлі:

$$Z_\Pi = \sqrt{15,1^2 + 0,11^2} \approx 15,1 \text{ (Ом)}.$$

Струм однофазного короткого замикання рівний:

$$I_k = \frac{220}{15,1 + 0,1} = 14,47 \text{ (A)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$I_k \geq k * I_n, \quad (4.3)$$

де I_n - номінальний струм спрацьовування плавкої вставки, А;

k - коефіцієнт кратності нелінійного струму I_n , А.

Коефіцієнт кратності нелінійного струму I_n розраховується по формулі (4.4.):

$$I_n = P / U, \quad (4.4)$$

де $P = 220$ Вт - споживана потужність;

$U = 220$ В - робоча напруга;

$k = 3$ А - для плавких вставок.

Отже, $I_n = 220 / 220 = 1$ А.

Підставивши значення у вираз (4.3), одержимо:

$$14,47 > 3 * 1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування(і захист) при підвищенні номінального струму.

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються [11], СНіП, відповідними ГОСТами і ОСТАми з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами. Мікроклімат виробничих приміщень [12] визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до [13] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні. За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу (категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл. 4.1).

Таблиця 4.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22...24	40...60	0,1
Тепло	23...25	40...60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції(з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й устанавленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створінню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної кількості свіжого повітря, визначеного в СНіП (30 м³ в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибирати найбільш шумозахисні матричні принтери або з великою швидкістю роботи(струменеві, лазерні). Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання(під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплону).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання : обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту [14].

Стандарт [14] пред'являє жорсткі вимоги в таких областях: ергономіка(фізична, візуальна і зручність користування), енергія, випромінювання(електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [15].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до [16] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(E_n) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛП02П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки $a=12$ м, шириною $b=10$ м, заввишки $z=4$ м, використовуючи формулу (4.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.5)$$

де F - світловий потік = 3120 лм;

E - максимально допустима освітленість робочих поверхонь = 200 лк;

S - площа підлоги = 120 м²;

Z - поправочний коефіцієнт світильника = 1,2;

k - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

n - кількість світильників;

U - коефіцієнт використання освітлювальної установки = 0,6;

M - кількість ламп у світильнику = 2.

З формули (4.5) виразимо n (4.6) і визначимо кількість світильників для даного приміщення:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.6)$$

Отже, $n = (200 \cdot 120 \cdot 1,2 \cdot 1,5) / (3120 \cdot 0,6 \cdot 2) = 12$.

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розмішувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 4.1.

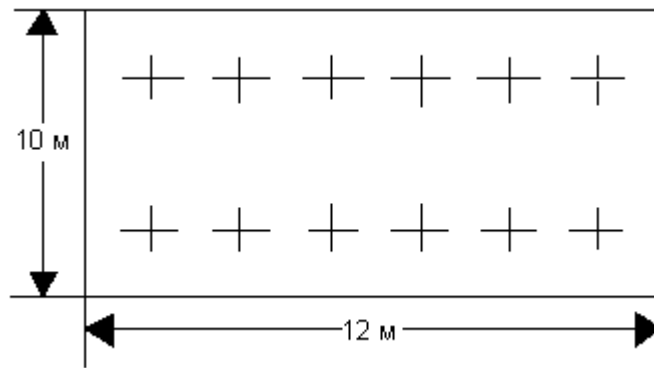


Рисунок 4.1 - Схема розташування світильників

4.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

- поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 С ;

- полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 С, температура самозаймання 530 С, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;

- стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;

- пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;

- деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 З, схильна до самозаймання [17].

Згідно [18] приміщення відносяться до категорії В(пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогнебезпечної зони класу П - Па (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегріву від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є : оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акролеїн, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза з коробкою марки В(жовта).

Пожежна безпека об'єктів народного господарства регламентується [19] і забезпечується системами запобігання пожежам і протипожежному захисту. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

- застосування устаткування, що задовольняє вимогам електростатичної безпеки;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;
- виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;

- підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з палим середовищем, нижче гранично допустимої, становить 80% якнайменшої температури самозаймання пального.

- заміна небезпечних технологічних операцій більш безпечними;
- ізольоване розташування небезпечних технологічних установок і устаткування;

- зменшення кількості палих і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;

- запобігання можливості утворення палих сумішей на лінії, вентиляційних системах і ін.;

- механізація, автоматизація та справність(поточкова) виробництва;

- суворе дотримання стандартів і точне виконання встановленого технологічного режиму;

- запобігання можливості появи в небезпечних місцях джерел запалення;

- запобігання розповсюдженню пожеж і вибухів;

- використання устаткування і пристроїв, при роботі яких не виникає джерел запалення;

- виконання вимог сумісного зберігання речовин і матеріалів;

- наявність громовідводу;

- організація автоматичного контролю параметрів, що визначають джерела запалення;

- ліквідація можливості самозаймання речовин і матеріалів .

- Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог :

- електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження і кондиціонування;

– система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;

– робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до [19] приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1(димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для ліквідації пожежі використовують вогнегасники (хімічно-пінні, пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар(сокири, ломи, багри, шерстяну або азбестову ковдри) [20]. Як засіб індивідуального захисту проектом передбачається використання промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

У розділі «Охорона праці» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки, заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці.

ВИСНОВКИ

У дипломній роботі бакалавра була поставлена задача створення додатку для мобільних пристроїв, працюючих на платформі Android, котре допомагало би проводити змагання з підводного плавання. Основною ідеєю було створення легкого для застосування додатку, що буде відображати дані з проведення змагань. Для цього були вирішені наступні завдання:

- була досліджена задача проведення змагань
- вирішено проблему швидкодії

У ході розробки були отримані навички проектування баз даних, програмування на PHP, програмування на Java під платформу Android.

Обрані технології:

- Java, SDK;
- MySQL;
- PHP.

Актуальність створеного додатка у тому, що аналогічних йому немає. У майбутньому цей додаток вдосконалиться та матиме більшу функціональність.

У розділі охорони праці були розглянуті умови роботи в офісі, шкідливі фактори на робочому місці. Приведені норми безпеки та необхідність інструктажу перед допуском до роботи. Був проведений розрахунок одностороннього природнього освітлення на робочому місці. Також були приведені норми пожежної безпеки для даного приміщення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1) Пишем своё первое приложение на Android [Электронный ресурс] - Режим доступа: www: URL: <http://habrahabr.ru/post/109944/>. - 28.04.2018 - Загл. з екрану.
- 2) Голощапов, А.Л. Google Android программирование для мобильных устройств [Текст] / А.Л. Голощапов - Санкт-Петербург .: BHV Санкт-Петербург, 2011 - 1549 с.
- 3) Хашими, С. Разработка приложений для Android [Текст] / С. Хашими. - Москва .: Бином, 2011. - 2125 с.
- 4) Дерси, Л. Android за 24 часа. Программирование приложений под операционную систему Google [Текст] / Л. Дерси. - Род Групп, 2011. - 1499с.
- 5) Rick Rogers John Lombardo, Android Application Development [Text] / Rick Rogers John Lombardo; - ECOM Pablysherz, ISBN 978-5-9790-0113-5, 978-0-596-52147-9;
- 6) Industry 4.0: На пороге промышленной революции [Электронный ресурс] - Режим доступа: www/ URL: <http://www.infocity.az/?p=17479>. - 18.05.2018 - Загл. з екрану.
- 7) Державний стандарт України. ГОСТ 12.1.005-88. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
- 8) Державний стандарт України. ГОСТ 12.1.005-88. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
- 9) Державний стандарт України. ГОСТ 12.0.003-74 Опасные и вредные производственные факторы. Классификация.
- 10) Нормативно-правові акти з охорони праці. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів
- 11) Державний стандарт України. ГОСТ 12.1.009-76. ССБТ. Электробезопасность. Термины и определения.
- 12) Державні санітарні норми України. ДСП 173-96. Державні санітарні правила планування та забудови населених пунктів.

13) Державні санітарні норми України. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень.

14) Державний стандарт України. ГОСТ 12.1.005-88. Система стандартів безпеки праці. Общие санитарно-гигиенические требования к воздуху рабочей зоны.

15) TCO' 07 Certified Displays. © 2007 Copyright TCO Development AB

16) Державні санітарні норми і правила. ДСанПіН 3.3.2.007-98, Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.

17) Державні будівельні норми ДБН В.2.5-28-2006. Природне і штучне освітлення

18) Державний стандарт України. ГОСТ 12.1.044-89 Система стандартів безпеки праці. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения.

19) Нормативні акти пожежної безпеки. НАПБ Б.03.002-2007. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою.

20) Державний стандарт України. ГОСТ 12.1.004-91. "Система стандартів безпеки праці. Пожарная безопасность. Общие требования".

21) Нормативні акти пожежної безпеки. НАПБ А.01.001-2014 "Правила пожежної безпеки в Україні"

22) Нормативні акти пожежної безпеки. НАПБ Б.03.001-2004. Про затвердження Типових норм належності вогнегасників.

ДОДАТОК А.

Лістинг коду

```

<?php namespace App\Controller;

use App\Model\Event;
use App\Model\Mapper\EventsMapper;
use Silex\Application;
use Symfony\Component\HttpFoundation\Request;

class Events
{
    public function getList(Application $app)
    {
        $eventsMapper = new EventsMapper($app);
        $events       = $eventsMapper->getAll();

        foreach ($events as &$event) {
            if ($event instanceof Event) {
                $event = $event->toArray();
            } else {
                $event = [];
            }
        }

        return $app->json($events);
    }

    public function getOne(Application $app, Request $request)
    {
        $eventsMapper = new EventsMapper($app);
        $event        = $eventsMapper->get($request-
>get('event_id'));

        if ($event instanceof Event) {
            return $app->json($event->toArray());
        } else {
            return $app->json([]);
        }
    }
}

<?php namespace App\Auth;

use App\Model\Mapper\CoachesMapper;
use Silex\Application;
use Silex\ServiceProviderInterface;
use Symfony\Component\HttpKernel\Exception\HttpException;

class SecurityProvider implements ServiceProviderInterface

```

```

{
    /**
     * @param Application $app
     */
    public function register(Application $app)
    {
        if (!isset($app['security.firewalls'])) {
            $app['security.firewalls'] = [];
        }

        $app->before(function () use ($app) {
            /** @var \Symfony\Component\HttpFoundation\Request
             $request */
            $request = $app['request'];

            $options = null;
            foreach ($app['security.firewalls'] as $n => $o) {
                if (isset($o['pattern']) &&
preg_match($o['pattern'], $request->getPathInfo())) {
                    $options = $o;
                    break;
                } elseif (isset($o['route'])) {
                    if (is_string($o['route']) && $o['route'] ==
$request->get('_route')) {
                        $options = $o;
                        break;
                    } elseif (is_array($o['route']) &&
in_array($request->get('_route'), $o['route'])) {
                        $options = $o;
                        break;
                    }
                }
            }

            if (isset($options['role_required']) &&
$options['role_required']) {
                $token = $request->headers->get('X-Bearer-
Token');

                if (!$token) {
                    throw new HttpException(401, 'Unauthorized',
null, [], 401);
                }

                /** @var \App\Auth\JsonWebToken\Service $jwt */
                $jwt = $app['jwt'];

                try {
                    $payload = $jwt->decodeToken($token);
                } catch (\ExpiredException $ex) {
                    throw new HttpException(401, $ex-
>getMessage(), $ex, [], 401);
                } catch (\Exception $ex) {

```

```

        throw new HttpException(400, $sex-
>getMessage(), $sex, [], 400);
    }

    if (!$payload || !isset($payload['id'])) {
        throw new HttpException(401, 'Unauthorized',
null, [], 401);
    }

    $coachesMapper = new CoachesMapper($app);
    $coach          = $coachesMapper-
>get($payload['id']);

    if (!$coach) {
        throw new HttpException(401, 'Unauthorized',
null, [], 401);
    }
}

return;
});
}

/**
 * @param Application $app
 */
public function boot(Application $app)
{
}
}
}

```

ДОДАТОК Б. Електронні плакати

Міністерство освіти і науки України
Східноукраїнський національний університет імені В.Дала

Дипломна робота бакалавра

Мобільний додаток реєстрації спортивних змагань

Ст. групи КІ-14з

Поддубко С.В.

Постановка задачі

Необхідно створити додаток під Android, за допомогою якого можна буде проводити змагання з підводного плавання. Ця програма має дозволяти заходити в систему управління під своїм логіном і вибирати змагання, інформацію про яких ви хочете подивитися, після чого повинен з'являтися список днів змагань і його видів. За допомогою програми можна спостерігати за стартовим протоколом і результуючим, вибираючи номер запливу. Для випадку, коли інтернет недоступний, необхідно створити внутрішню базу даних.

Актуальність даного додатка

- ▶ Актуальність даної програми полягає в тому, що аналогічних програм не існує. Додаток затребуване, тому що безліч людей має бажання ознайомитись з результатами змагань з підводного плавання, але для цього потрібно шукати спеціалізовані сайти з оновленою інформацією, а для цієї програми дозволяє спостерігати за цим мобільно, тобто за допомогою мобільного пристрою на платформі Android.

Технології

- ✓ Мова Java - об'єктно-орієнтована мова програмування
- ✓ Мова PHP - скриптова мова програмування
- ✓ HTTP бібліотека Volley Framework
- ✓ Restfull architecture - стиль архітектури програмного забезпечення для побудови веб-служб

Работа програми



Работа програми



Работа программы

	Стартовый уровень	Результатный уровень
1	Он Карбонил Ирида ДЮССА Акаландар - 1274 и Кале	0.36.00
2	Он Чарлтон Дарк Даркиса 303 и Кале	0.02.00
3	Он Салмон Дарк Даркиса 303 и Кале	0.25.00
4	Он Ичелон Владислав Даркиса 303 и Кале	0.17.00
5	Он Кацин Катрина ДЮССА Акаландар - 1274 и Кале	0.35.00
6		

Работа программы

	Стартовый уровень	Результатный уровень
1	Он Карбонил Ирида ДЮССА Акаландар - 1274 и Кале	0.36.00
2	Он Чарлтон Дарк Даркиса 303 и Кале	0.02.00
3	Он Салмон Дарк Даркиса 303 и Кале	0.25.00
4	Он Ичелон Владислав Даркиса 303 и Кале	0.17.00
5	Он Кацин Катрина ДЮССА Акаландар - 1274 и Кале	0.35.00
6		

- Запись 1
- Запись 2
- Запись 3
- Запись 4
- Запись 5
- Запись 6
- Запись 7

Висновки

У кваліфікаційній випускній роботі бакалавра була поставлена задача створення програми для мобільних пристроїв на платформі Android, яке дозволяло б стежити за інформацією про проведення змагань з підводного плавання. Основною ідеєю було створення простого, легкого для використання, швидкого додатки.

Доповідь закінчено, дякую за увагу.