

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

НКЦ заочної освіти

Кафедра інформаційних технологій

Бакалаврська кваліфікаційна робота

на тему: Розробка мобільного застосування для визначення мінералів

Виконав студент 5 курсу групи КН-5
Напряму 6.050101, комп'ютерні науки
Іжиков Владислав Олегович

Керівник ст.викладач
Вохменцева Тетяна Борисівна

Консультант д.т.н., професор
Мещеряков Володимир Іванович

Рецензент к.ф.-м.н., доцент
Ткач Тетяна Борисівна

ЗМІСТ

Вступ.....	5
1 Сучасні підходи до класифікації природніх об'єктів	7
1.1 Таксономія	7
1.2 Таксономічні категорії.....	15
1.3 Визначення як завдання класифікації. Ключі визначення.....	16
1.4 Друковані визначники	18
1.5 Приклади електронних визначників	19
1.5.1 Каталог-визначник мінералів і гірських порід.....	19
1.5.2 Визначник мінералів онлайн.....	20
1.5.3 Визначник мінералів та гірських порід.....	21
1.6 Постановка завдання.....	22
2 Проектування системи.....	23
2.1 Призначення програмного комплексу	23
2.2 Загальна архітектура системи	23
2.3 Обґрунтування вибору програмних засобів реалізації системи.....	26
2.4 Проектування інформаційної системи за допомогою методології функціонального моделювання	35
2.5 Специфікація вимог до бази даних	42
2.6 Побудова моделі даних.....	43
2.7 Побудова фізичної моделі даних.....	46
3 Реалізація системи.....	47
3.1 Структура та компоненти визначника	47
3.2 Організація інтерфейсу.....	49
3.3 Дизайн системи	49
3.4 Опис роботи з програмою	50
Висновки	55
Перелік джерел посилання	56

ВСТУП

Актуальність теми обумовлена необхідністю розробки програмного забезпечення, яке здійснює пошук та перегляд природних об'єктів за декількома критеріями. Завданням інформаційного пошуку є знаходження відповідних (до пошукового запиту) інформаційних об'єктів, або документів серед доступного для пошуку матеріалу. Пошукова система переглядає всі доступні інформаційні одиниці зі збірки і відбирає документи відповідні до інформаційного запиту.

Важливим поняттям при роботі з інформацією є класифікація об'єктів – система розподілу об'єктів (предметів, явищ, процесів, понять) по класах відповідно до певної ознаки. Класифікація повинна фіксувати закономірні зв'язки між класами об'єктів з метою визначення місця об'єкту в системі, яке вказує на його властивості. Класифікація об'єктів – це процедура угруповання на якісному рівні, направлена на виділення однорідних властивостей. У цьому аспекті класифікація служить засобом зберігання і пошуку інформації, що міститься в ній самій. Іншим завданням класифікації є проведення ефективного пошуку інформації або яких-небудь об'єктів, що містяться в спеціальних сховищах.

Об'єкт дослідження. Практично у кожній галузі людської діяльності використовуються ті або інші класифікатори. Систематичні бібліотечні класифікації, багаточисельні класифікації товарів, класифікації галузей знання, класифікації об'єктів інтелектуальної власності, класифікації виробів промисловості (верстатів, приладів, деталей, інструменту, і так далі), класифікації посад і звань, класифікації нормативних актів, класифікація хімічних сполук, класифікація рослин, тварин, ґрунтів, мінералів тощо.

Предмет дослідження. Важливим завданням класифікації є діагностика (визначення, тобто знаходження місця організму в системі). Під діагностикою розуміють передусім складання таблиць для визначення організмів, так званих визначників. Визначники широко застосовуються при біологічних досліджен-

нях, в практиці екологів, при роботі з гербарієм, складанні флористичних описів і т.д., проте, існуючі визначники, через консервативність свого виконання, вже не можуть відповідати параметрам, що пред'являються сучасною реальністю. В зв'язку зі швидким зростанням об'єму інформації виникає гостра необхідність розробки і впровадження автоматизованих інформаційних систем з використанням новітніх методів обробки даних.

Мета і завдання дослідження. Головною метою даної роботи є проектування та створення такої електронної інтерактивної системи класифікації та діагностики мінералів, яка могла би використовуватися лише за рахунок незначних змін в інтерфейсі та наповнення бази даних, або незначних змін у самій базі.

Електронні визначники можуть бути використані в учбовому процесі. Вони можуть використовуватися як в демонстраційному режимі при вивченні нового матеріалу або повторенні і узагальненні пройденого, так і в режимі виконання практичних робіт, що виконуються у комп'ютерному класі.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- провести дослідження предметної області;
- розглянути методи класифікації та побудови ключів визначення;
- розглянути приклади друкованих визначників для різних природних об'єктів;
- виконати огляд існуючих електронних аналогів;
- провести проектування функціональної моделі і вибрати апаратну платформу для її реалізації;
- провести проектування бази даних інформаційної системи;
- провести проектування застосування;
- обрати інструментальне середовище розробки і виконати програмну реалізацію інформаційної системи.

Пояснювальна записка містить 56 сторінок, 22 рисунки, 13 посилань.

1 СУЧАСНІ ПІДХОДИ ДО КЛАСИФІКАЦІЇ ПРИРОДНИХ ОБ'ЄКТІВ

1.1 Поняття та приклади таксономії

Таксономія – ієрархічно збудована система цілей і результатів від простої до складної системи. Побудова таксономій задає перспективні системи тестування різних сфер діяльності людини: когнітивну, афективну, психомоторну та інші.

Математично таксономією є деревоподібна структура класифікацій певного набору об'єктів. Зверху цієї структури об'єднуюча єдина класифікація – кореневий таксон – яка відноситься до всіх об'єктів даної таксономії. Таксони, що знаходяться нижче кореневого, є більш специфічними класифікаціями, що відносяться до піднаборів загального набору об'єктів, що класифікуються. Нижче в цій таксономії знаходяться тип, клас, загін, сімейство, рід і вид.

Зараз під терміном таксономія розуміється тип керованого словника, що має ієрархічну структуру. У таксономії конкретні (більш специфічні, дочірні) елементи входять в широкі (загальніші, батьківські) елементи.

Існують і такі ієрархії, що не пов'язані з об'єктами природи, наприклад, бібліотечно-бібліографічна класифікація (ББК) для організації бібліотечних фондів [1]¹⁾, систематичних каталогів та картотек.

За допомогою ББК працівники бібліотек можуть визначити, до якого розділу належить книга, не читаючи її. Фрагмент з класифікатору ББК показано на рис. 1.

Систематика мінералів, як і будь-яка інша систематика, повинна базуватися на спільності всіх ознак, або тих з них, які є сутнісними і з якими всі інші ознаки знаходяться в певній залежності.

¹⁾ [1] Библиотечно-библиографическая классификация для организации библиотечных фондов. URL:<http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2000/part2/SI> (дата звернення 24.02.2019).

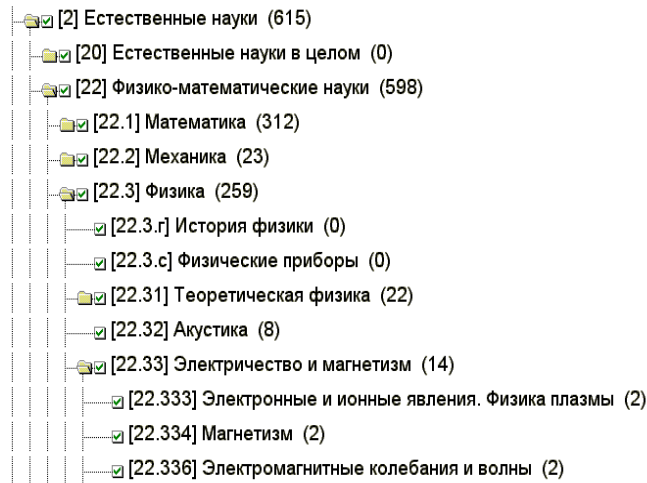


Рисунок 1 – Приклад ієрархії елементів ББК

Мінеральний вид – основна таксономічна одиниця сучасної мінералогії. З погляду логіки, вид у сучасній мінералогії – це сукупність мінеральних індивідів з певними ознаками схожості, за якими вони водночас відрізняються від решти індивідів. Точне сучасне визначення: мінеральний вид – сукупність мінеральних індивідів, що належать до однієї просторової групи симетрії й характеризуються однаковим (близьким) хімічним складом або безперервною зміною складу в природних межах [2]¹⁾.

Приклад ієрархії мінералів представлено на рис. 2.



Рисунок 2 – Приклад ієрархії мінералів

¹⁾ [2] Павлишин В. Вступ до мінералогії (курс лекцій). К.:Вид-во ДГЦУ, 1997. 80 с.

Кожен мінерал характеризується певним хімічним складом і має характерну для нього внутрішню будову. Ці дві важливі особливості обумовлюють досить постійні та індивідуальні фізичні властивості мінералів.

У кожного мінералу є свої, властиві лише йому ознаки. Для одних мінералів постійною ознакою є колір, для інших твердість, для третіх щільність, для четвертих форма кристалів і т.д.

При визначенні мінералів за зовнішніми ознаками необхідно звертати увагу в першу чергу на загальні для всіх мінералів ознаки, а потім переходити до розгляду індивідуальних особливостей окремих мінералів.

У першу чергу звертається увага на блиск мінералів, після цього на твердість, на колір, на риску і т.д.

Блиск. Мінерали бувають блискучі і матові – коли блиск у мінералів відсутній, наприклад у піролюзиті. Більшість мінералів має здатність відбивати від своєї поверхні світло, що й обумовлює блиск мінералів.

Мінерали по блиску легко діляться на дві групи: ті, що мають металевий блиск і ті, що мають неметалевий блиск.

Блиск металевий. Металевий блиск нагадує блиск поверхні свіжого зламу металів. Його добре видно на неокислених поверхнях зразків. Мінерали, з металевим блиском, непрозорі і більш важкі порівняно з мінералами, що мають неметалевий блиск. Іноді завдяки процесам окислення мінерали, що мають металевий блиск, покриваються матовою кіркою.

Металовидний блиск – більш тусклий, як у металів, що стали тускими від часу. Металовидний блиск спостерігається у мінералів, які також є рудами різних металів (залізіста цинкова обманка, бурий залізняк, магнітний залізняк).

Блиск неметалічний. Неметалічний блиск може бути різним: скляним, алмазним, перламутровим, шовковистим, жирним, восковим.

Велика частина мінералів має неметалічний (переважно скляний) блиск. Набагато менше мінералів з металевим блиском. Блиск необхідно спостерігати на свіжому зламі мінералу. При визначенні блиску колір мінералу не приймається до уваги.

Твердість. Твердість – це опір твердого тіла руйнуванню в поверхневому шарі при силовому впливі на нього. Якщо випробуваний мінерал м'якше, ніж той предмет або мінерал, яким ви дряпаєте по його поверхні, то на ньому залишається слід-подряпина.

Твердість мінералів залежить від характеру сил зчеплення між частинками речовини, що визначається формою кристалічної решітки, тобто взаємним розташуванням частинок. Якщо енергетичний зв'язок між частинками сильніший, тоді і твердість вища.

Кристалічні речовини, у яких будова відрізняється в різних напрямках, мають і різну твердість у цих напрямках. По твердості мінерали можна розділити на чотири групи: м'які мінерали, мінерали середньої твердості, тверді мінерали, дуже тверді мінерали.

Для визначення твердості мінералу необхідно вибрати чисті ділянки (можуть бути присутніми у невеликій кількості інші мінерали). Після випробування треба стерти порошок з поверхні зразка, тобто роздроблені частки, і переконатися, що на зразку справді залишився слід (подряпина), оскільки порошок міг утворитися з того предмета, яким дряпали мінерал.

Колір. Тверді і дуже тверді мінерали в подальшому діляться за кольором. Колір у мінералів буває найрізноманітніший. Для деяких мінералів колір є постійною ознакою.

Для більшості мінералів ця ознака непостійна. Польові шпати бувають білого, жовтого, червоного, зеленого, темно-сірого кольорів. Кальцит зустрічається безкольоровий, білий, жовтий, зелений, блакитний, фіолетовий, бурий, чорний. Тому не слід визначати мінерали тільки за кольором, завжди потрібне додаткове визначення інших ознак.

Колір мінералу залежить від їх хімічного складу, від наявності сторонніх домішок, від стану атомів і іонів всередині кристалу, від розсіювання променів світла всередині мінералу, від інтерференції і дифракції світлових хвиль.

Забарвлення мінералів визначається в першу чергу їх хімічним складом. Кожен хімічний елемент, що входить до складу мінералів, і кожна хімічна сполука надає їм певне забарвлення. Наявність навіть мізерної кількості домішок вистачає, щоб мінерал отримав нове забарвлення.

Для визначення кольору мінералів необхідно отримати свіжий злам. За кольором мінерали поділяються на шість груп:

- колір білий, сіруватий або мінерал безбарвний;
- колір жовтий, бурий, коричневий, рожевий, червоний;
- колір зелений;
- колір блакитний, синій, фіолетовий;
- колір темно-сірий, чорний;
- забарвлення мінералу строкате, багатобарвне, зональне.

Риска мінералів. М'які мінерали і мінерали середньої твердості діляться за кольором риси.

Колір порошку у деяких мінералів не відрізняється від кольору самого мінералу, але зустрічаються й такі мінерали, колір порошку яких різко відрізняється від кольору мінералу, і в такому випадку це має важливе значення при визначенні. Наприклад, у мінералу піриту колір-світлий латунно-жовтий, порошок - чорний зі слабким зеленуватим відтінком. Кальцит буває безбарвний, білий, жовтий, зелений, блакитний, синій, фіолетовий, чорний. Порошок у кальциту білий, незалежно від кольору мінералу. Для одержання порошку мінералу застосовується матова фарфорова пластинка, так званий "бісквіт". Якщо провести мінералом по поверхні бісквіту, мінерал залишає на її поверхні слід (риску).

Більшість твердих мінералів риси не дає, дуже тверді мінерали (усі без винятку) риси не дають. Ці мінерали можуть дряпати бісквіт і створювати

уявлення ризику. Можна вважати, що мінерал дає ризику, якщо вона стирається пальцем.

Бісквіт можна замінити осколком фарфорового посуду знявши з нього наждачним папером або напилком гладкий шар глазури. У разі відсутності фарфорового посуду досить подряпати мінерал ножом і отримати тонкий порошок.

Для визначення кольору ризику необхідно цей порошок розмазати на білому папері. За кольором ризику виділяються п'ять груп:

- ризика біла або ризику не дає;
- ризика жовта, помаранчева, бура, коричнева, червона;
- ризика блакитна, синя, фіолетова;
- ризика зелена;
- ризика сіра до чорної.

Побіжалість. Деякі мінерали, особливо ті, які містять мідь, на своїй поверхні мають різнокольорову тонку плівку: рожеву, червонувату, жовтувату, блакитну та ін., обумовлену процесами хімічного вивітрювання. Колір цієї плівки відрізняється від кольору самого мінералу. Це явище отримало назву побіжалості. Вона особливо характерна для халькопіриту. У халькопіриту колір латунно-жовтий. На поверхні халькопіриту нерідко в результаті хімічної реакції утворюється плівка райдужного або синього кольору. Побіжалість спостерігається тільки у мінералів з металевим блиском.

Горючість і плавкість мінералів. М'які і середньої твердості мінерали, які мають неметалевий блиск, надалі поділяються на дві групи: на мінерали, які легко плавляться або горять, і на тугоплавкі і негорючі мінерали.

До горючих та легкоплавких мінералів можна віднести сірку самородну. Одні мінерали плавляться і спалахують від сірника, інші – від свічки. При горінні мінералів виділяються гази різного складу, які мають різний запах. Так, наприклад, сірку самородну іноді буває важко відрізнити від бурштину по зовнішнім ознакам. Обидва мінерали характеризуються легко-

плавкістю і здатністю горіти. Відрізняються вони за запахом газів, що утворюються при горінні: самородна сірка виділяє запах різкий, задущливий; янтар – приємний, ароматичний.

Якщо з'явиться необхідність досліджувати мінерал на горіння або плавкість, слід відколоти від нього маленький шматочок, затиснути його кінчиком пінцета і ввести в полум'я свічки, спиртівки або газового пальника.

Розчинність мінералів у воді. Мінерали відрізняються різним ступенем розчинності. Деякі з них досить легко розчиняються у воді, такі як кам'яна сіль, силвін, карналіт, мірабіліт. Більшість мінералів, легко розчиняються у воді, мають смак і за цією ознакою легко відрізняються від інших мінералів. Деякі гірше розчиняються у воді гіпс, ангідрит, кальцит, доломіт, магнезит. Вони не мають смакових якостей. Інші фундаментальні мінерали розчиняються у воді погано або практично не розчиняються у воді. У групі м'яких і середньої твердості мінералів, що володіють неметалевим блиском і мають білу риску, можна виділити мінерали, що відрізняються смаковими якостями (наприклад, кухонна сіль).

Розчинність мінералів в кислотах. Деякі мінерали, що мають у своєму складі вуглекислі солі, під дією соляної кислоти (10 % розчин) виділяють у вигляді бульбашок вуглекислий газ (таку ж дію надає оцтова кислота).

Після того як проведено поділ на групи по блиску, твердості, по рисці, за кольором і т.д., всередині групи звертається увага на спайність, злам, щільність.

Спайність. Спайність виражається в тому, що в певних напрямках мінерали мають меншу здатність опиратися зовнішньому фізичному впливу – в цьому напрямку вони легше розколюються і дають рівні, гладкі, блискучі поверхні спайності.

Спайність – одна з найцікавіших особливостей кристалічних мінералів: вона обумовлена закономірним розташуванням атомів і іонів всередині кристалів і пояснюється тим, що в просторовій решітці існують плоскі сітки,

тяжіння між якими найменше, внаслідок, наприклад, більшої відстані між системами атомів або іонів.

Злам. При розколі у мінералів виникають поверхні, які визначають так званий злам. Мінерали, які мають спайність, дають рівний злам, наприклад кальцит. Мінерали, позбавлені спайності, мають нерівний злам, наприклад кварц. Злам, схожий на поверхню раковини, отримав назву раковистого. Раковистий злам буває у халцедону. Зернистий злам характерний для піриту та інших мінералів, які мають зернисту будову. У деяких мінералів злам землистий.

Щільність. Щільність у мінералів буває різна і залежить від їх хімічного складу. Мінерали, до складу яких входять такі важкі елементи, як свинець, вольфрам, барій тощо, мають велику щільність (важкі), а мінерали, до складу яких входять легкі елементи, наприклад, алюміній, калій, натрій і т.п., мають невелику щільність (легкі). Найбільш важкими є самородні метали.

При визначенні мінералів за зовнішніми ознаками щільність з великою точністю не визначається. При цьому досить розподілу мінералів на дві групи: легкі і важкі, причому необхідно розрізняти легкі і важкі серед мінералів, які мають металевий блиск і в групі з неметалічним блиском.

Ковкість і крихкість. Ковкі мінерали при ударі молотком сплющуються і закруглюються по краях, в той час як крихкі при ударі розсипаються на дрібні шматки. При дряпанні ножом крихких мінералів утворюється порошок, при дряпанні ковких – порошку не утворюється і на поверхні залишається блискучий слід. Серед мінералів з неметалічним блиском можна виділити крихкі, які легко розсипаються, і в'язкі.

Магнітність. Магнітність мають мінерали, що містять залізо (магнітний залізняк та ін.) Для визначення магнітних мінералів користуються магнітною стрілкою, а в польових умовах роботи - стрілкою компаса. Мінерали, що мають магнітні властивості, при наближенні їх до магнітної стрілки притягують останню або відштовхують.

1.2 Таксономічні категорії

Систематика виробила свою систему понять і символів, свою мову, що служить для класифікації організмів. Будь-яка система класифікації є немичуче системою ієрархічно супідрядних одиниць. Така система взаємопідлеглих груп є єдиним логічно можливим засобом розташування організмів в певній впорядкованій системі. Тому кожна система класифікації підрозділяється на визначені, супідрядні один одному систематичні категорії, або одиниці. Для позначення систематичних одиниць будь-якого рангу прийнятий термін «таксон».

Хімічне та структурне (кристалохімічне) розмаїття мінералів твориться в природі за рахунок прояву законів поєднання (зв'язування) атомів і явища ізоморфізму. Кристалічна структура мінералів підпорядковується 230 просторовим групам симетрії. Найголовнішими типами хімічного зв'язку в мінералах є йонний, ковалентний, металічний й молекулярний. З погляду розподілу хімічних зв'язків мінерали поділяються на ізодесмічні та гетеродесмічні. Серед перших можна виділити мінерали з координаційним і каркасним мотивами, серед других – мінерали з острівним, ланцюжковим і шаруватим мотивами. Структурний мотив – це якісна структурна характеристика мінералів, яка відбиває найістотніші риси їх атомної будови, яскраво-адекватно віддзеркалюється в морфології та фізичних властивостях мінеральних кристалів.

Найважливішою в мінералогії є кристалохімічна класифікація мінералів, в якій зазвичай виділяють такі таксони: царство, тип, клас, підклас, група, вид, різновид. У земній корі найпоширеніші мінерали класу силікатів та алюмосилікатів. За ними йдуть оксиди, гідроксиди, карбонати. Незначна частина земної кори представлена простими речовинами, сульфідами, галогенідами, іншими класами [3]¹⁾.

¹⁾ [3] Поваренных А. С. Кристаллохимическая классификация минеральных видов. К.: Наук. думка, 1966.132 с.

Форма й розміри мінералів дуже розмаїті – від видимих лише в електронному мікроскопі до кристалів-гігантів об'ємом декілька десятків кубічних метрів і масою в десятки й навіть сотні тонн. Фізичні властивості мінералів є функцією їхньої конституції (взаємопов'язаних хімічного складу та кристалічної структури). Найважливішими з них є: густина, оптичні (у тім числі забарвлення), механічні, магнітні, електричні, люмінесцентні властивості. За поширенням мінерали умовно поділяють на породоутворювальні, рудні або рудотвірні, другорядні, акцесорні, рідкісні й дуже рідкісні.

1.3 Визначення як завдання класифікації. Ключі визначення

Визначити природний об'єкт – означає знайти його наукову назву і систематичне положення, без чого неможливе пізнання світу природних об'єктів, побудова систем і класифікацій. Як правило, в завдання входить визначення до рангу вигляду. Для визначення природних об'єктів служать визначальні таблиці або ключі, по яких можна встановити назву потрібного природного об'єкта і його ранг. Ключі є сукупністю впорядкованих ознак і набором правил, що приписують певну послідовність виконання логічних операцій. Інколи «ключами» називаються і самі визначники.

Розрізняються три основні типи ключів: монотомічні (лінійні), дихотомічні і политомічні.

Монотомічні – найбільш примітивні: ключі будуються на принципі простого переліку. Монотомічний ключ передбачає створення таблиці, що містить майже повний опис природного об'єкта. Аналізуючи її, потрібно вирішити, чи відповідає опис визначуваному природному об'єкту. Це проста, найбільш стара форма визначення природних об'єктів. Істотною перевагою монотомічних ключів є їх простота. Групи ознак (і таксони відповідно) можуть розташовуватися в ключі в будь-якій послідовності (таксономічною, віковою, алфавітною і ін.). Це дозволяє необмежено розширювати об'єм визначника у міру

включення в нього нових таксонів. Недолік лінійних ключів – їх надзвичайна громіздкість.

Більшість діагностичних алгоритмів будуються в даний час за дихотомічним принципом. Дихотомічний ключ є широко відомим прийомом встановлення характеристики досліджуваного об'єкту по парах взаємовиключних характеристик, згрупованих за ознакою їх альтернативності (тези і антитези). Дихотомічні ключі зручні за наявності невеликої кількості діагностичних ознак і для визначення груп об'єктів, об'єм яких постійно залишається стабільним. При цьому обов'язковою умовою є повне збереження досліджуваного матеріалу.

Політомічний ключ передбачає відмову від традиційних тез і антитез і здійснення діагностики по сукупності комбінацій ознак[4]¹⁾.

Визначальні таблиці (ключі), будуються за ступінчасто-последовним принципом. На кожному рівні той, що визначає повинен вирішити, який з декількох подальших рівнів він повинен вибрати, аналізуючи ознаки визначуваного природного об'єкта.

Різновидом політомічних ключів є порівняльні таблиці, при користуванні якими можна оглянути відразу всю сукупність ознак кожного визначуваного роду або вигляду.

Важливою характеристикою визначника або ключа є число входів в ключ, тобто число ознак, з яких можна почати новий діагноз або черговий його крок.

Зазвичай виділяють однохідні ключі, в яких у користувача немає вибору, – як на першому кроці, так і на подальших, і він повинен користуватися єдиною пред'явленою йому ознакою; і багатохідні ключі, в яких на кожному кроці користувачеві надаються декілька ознак і він вибирає з них найбільш зручний і надійний.

Легко можна побудувати однохідний політомічний ключ і багатохідний дихотомічний.

¹⁾ [4] Балковский Б. Е. Цифровой политомический ключ для определения растений. К.: Наук. думка, 1964. 36 с.

1.4 Друковані визначники

Майже всі сучасні друковані визначники природних об'єктів побудовані по дихотомічній системі розділення рівнів.

Найчастіше у визначниках є окремі дихотомічні ключі для визначення сімейств, родів. Інколи вдається створити єдиний дихотомічний ключ для визначення таксонів різного рангу (рід, вид, підвид, група різновидів, різновид). З його допомогою, поступово відкидаючи або обираючи той або інший рівень, можна визначити до рангу виду або різновиду будь-яку рослину. Так, наприклад, побудований визначник «Определитель минералов» Азізова З.К. та Пьянкова С.А [5]¹⁾, взятий за основу в даній роботі. В ньому чітко визначені ознаки за якими відбувається класифікація мінералів та гірських порід. Також у цьому визначнику надані відомості про властивості породоутворюючих мінералів. Матеріал поданий в табличній формі, існує наявність ключа та алфавітного вказівника, що забезпечує швидку роботу з визначником.

У цьому посібнику зазначено, що він призначений для самостійної роботи студентів будівельних спеціальностей по визначенню основних мінералів. Може використовуватися при вивченні курсів з інженерної геології, матеріалознавства, на лабораторних заняттях з геологічного вивчення та польової геологічної практики.

Існують інші друковані визначники мінералів, наприклад: «Определитель горных пород» – автор Р. Юбельт [6]²⁾, «Определитель минералов, горных пород и окаменелостей» – автор В.Г.Музафарова [7]³⁾.

¹⁾ [5] Азизов З.К., Пьянков С.А. Определитель минералов: Учебное пособие. Ульяновск: УлГТУ, 2006. 53 с.

²⁾ [6] Юбельт Р. Определитель горных пород. М.:Мир, 1977. 260 с.

³⁾ [7] Музафаров В. Г. Определитель минералов, горных пород и окаменелостей. М.:Недра, 1977. 327 с.

1.5 Приклади електронних визначників

Перші роботи по використанню комп'ютерів для ідентифікації біологічних об'єктів з'явилися ще в кінці 60-х років минулого століття. На початку 70-х років спостерігався перший пік активності в області розробки методів комп'ютерної ідентифікації.

Але лише з широким поширенням персональних комп'ютерів стали з'являтися не лише теоретичні статті, але і дійсно зручні програми для діалогового визначення і автоматизованої побудови ключів визначення.

На даний час існує достатньо велика кількість електронних визначників різноманітної тематики. Серед них є і визначники мінералів.

1.5.1 Каталог-визначник мінералів і гірських порід

Розглянемо каталог-визначник мінералів і гірських порід, розміщений на сайті Бондарева Всеволода Петровича [8]¹⁾. Він призначений для широкого круга користувачів, як для аматорів, так і для професіоналів.

Визначник мінералів розрахований на користувачів з різним рівнем геологічних знань. На відміну від друкованих визначників і їх електронних клонів, цей визначник мінералів не використовує дихотомічний ключ.

Замість цього можна вказати довільне число ключових ознак і отримати в результаті набір видів, чії описи задовольняють умовам запиту (рис. 3).

Це інтерактивний каталог-визначник мінералів. Що стосується інтерфейсу цього каталогу – він не зовсім зручний для користувача, тому що всі характеристики мінералів не класифікують, хоча б, за зовнішніми і фізичними властивостями.

¹⁾ [8] Сайт Бондарева В.П. URL: <http://vsevolodbondarev.com/html/minerals.htm> (дата звернення 1.03.2019).

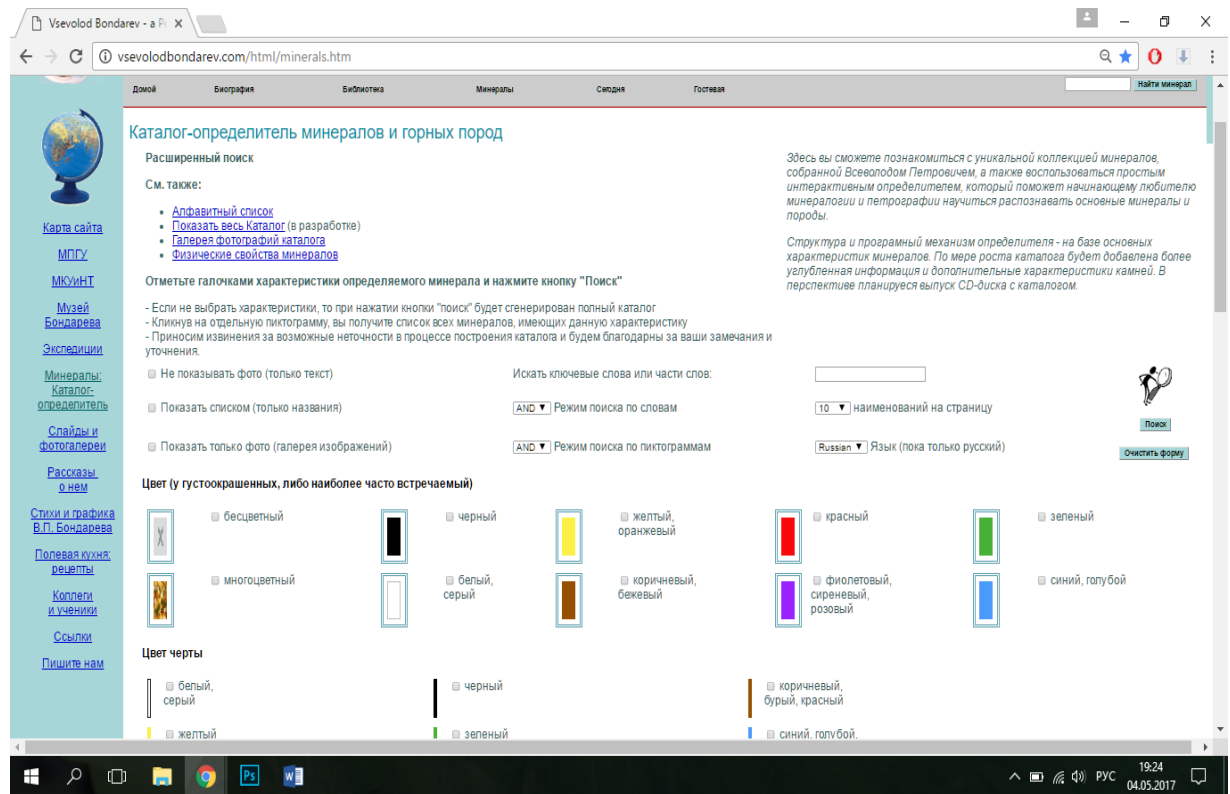


Рисунок 3 – Приклад задання ключа в системі

1.5.2 Визначник мінералів онлайн

Як і розглянутий каталог-визначник мінералів і гірських порід, сайт про мінерали, властивості каміннів допомагає користувачам визначати потрібний мінерал (рис.4). Також на цьому сайті можна знайти цікаві статті про мінерали, легенди про каміння, каміння для різних знаків Зодіаку та інше.

У розділі "Про сайт" написано, що сайт присвячений камінням, які виділяються своєю красою або володіють незвичайними цілющими або магічними властивостями [9]¹⁾.

Інтерфейс цього сайту не зовсім зручний у використанні. Також відсутня мобільна версія сайту, що ускладнює пошук мінералу з мобільного пристрою.

¹⁾ [9] Сайт о минералах, свойствах камней, легенды о камнях, магия камней. URL: <http://world-of-stones.ru> (дата звернення 2.03.2019).

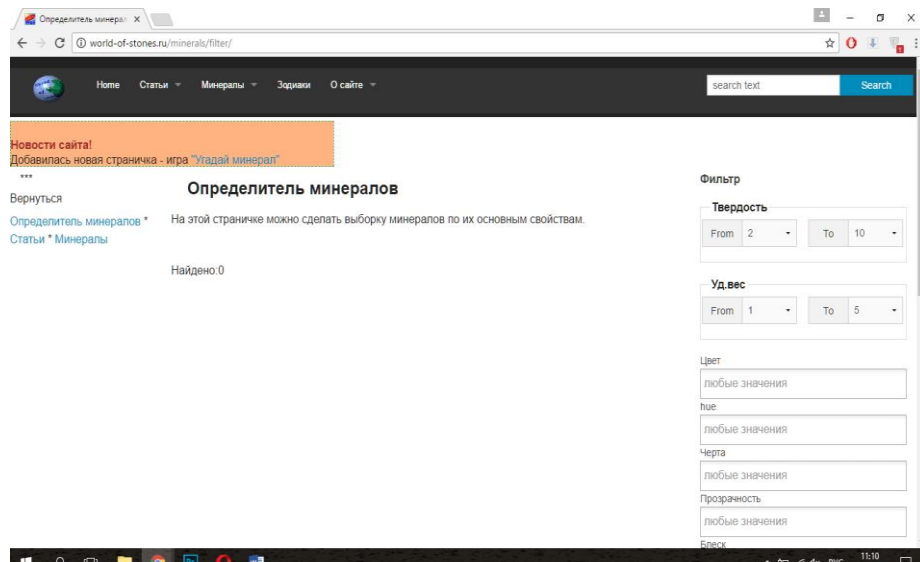


Рисунок 4 – Сторінка визначника мінералів

1.5.3 Визначник мінералів та гірських порід

Визначник мінералів та гірських порід складений для студентів, які вивчають геологію, також він може бути корисним для любителів мінералів та школярів. Визначник мінералів включає короткі діагностичні характеристики 94 мінералів з основних класів мінералів та їх кольорові фото. Приклад реалізації пошуку мінералу представлений на рис. 5.

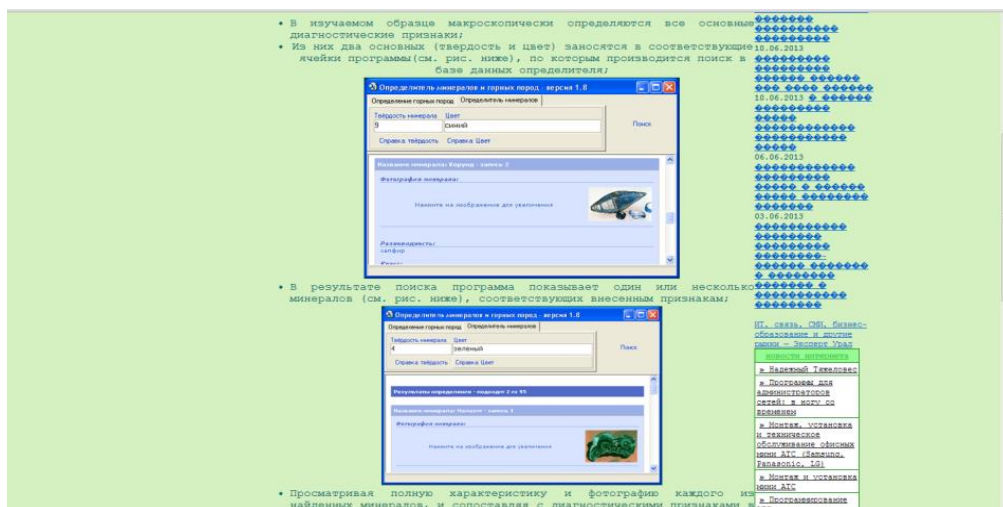


Рисунок 5 – Приклад реалізації пошуку мінералу

1.6 Постановка завдання

Кваліфіковане визначення таксономічної приналежності природних об'єктів є необхідною, але однією з самих трудомістких частин всіх досліджень.

В даний час кваліфіковане визначення таксономічної приналежності є майже мистецтвом, доступним, як правило, фахівцям з порівняно вузьких таксономічних груп. Для професійного визначення і, зрештою для верифікації досліджень, особливо важливе те, що інформація, що постійно з'являється, про зміни у визначальних таблицях, викликаного описом нового вигляду або ревізією таксонів, а також відомостями про їх ареали і біологічні особливості, зазвичай розсіяна по багаточисельних журнальних публікаціях і доступна лише вузькому колу фахівців. Збір і обробка цієї інформації вимагає значних витрат праці і часу.

Ці проблеми і труднощі можуть бути значною мірою усунені або істотно зменшені за рахунок використання можливостей сучасних інформаційних технологій, і, перш за все, засобів Інтернет.

Також на даний момент немає мобільних версій визначника мінералів, є лише веб-сайти, які не зовсім зручні при використанні на мобільному пристрої. Тому з'явилася необхідність розробити мобільний додаток, який буде зручно використовувати в польових умовах для визначення мінералів.

2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Призначення програмного комплексу

Кваліфіковане визначення таксономічної приналежності природних об'єктів є необхідним, але одним з самих трудомістких частин всіх досліджень в області біорізноманітності. Визначник є банком таксономічних знань, який можна широко використовувати не лише як дослідницький інструмент, але і як навчальний посібник.

Основний принцип, що закладається в систему, полягає в створенні зручних інтерфейсів додатків для роботи біологів, мінералогів із створення і актуалізації баз даних і роботи з мобільними визначниками природних об'єктів.

Технологія і програмне забезпечення дозволять:

- раціонально побудувати процес визначення природних об'єктів;
- істотно спростити процес визначення мінералів.

Спроектоване застосування є мобільним додатком, розробленим за допомогою web-мов HTML5, CSS і JavaScript. Також цей додаток легко переноситься між різними платформами.

2.2 Загальна архітектура системи

У технологічну основу створення системи покладені базові технології Internet, тобто інтегрована інформаційна система, що працює в мережі Інтернет на основі клієнт-серверних технологій.

Трирівневий додаток складається з наступних компонентів (рис.6).

Інтерфейсний (звичайно графічний) компонент представляє перший рівень. Це може бути, як віконний додаток, так і консольний клієнт. У багатьох випадках цей рівень може бути представлений застосуванням з Web-інтерфейсом. Перший рівень не повинен мати прямих зв'язків з базою даних і зберігати стан додатку.

На другому рівні розташовується сервер застосувань. Це може бути програмна система, написана на будь-якій сучасній мові програмування, але, як правило, найчастіше реалізація створюється на мовах Java, C# або C++. На другому рівні зосереджена велика частина прикладної логіки. Поза ним залишаються фрагменти, що експортуються на клієнтські системи, а також, занурені в третій рівень, такі, що зберігаються процедури і тригери.

Третій рівень забезпечує зберігання даних. Звичайно це стандартна реляційна або об'єктно-орієнтована СУБД. Якщо третій рівень є базою даних разом з процедурами, що зберігаються, тригерами і схемою, що описує застосування в термінах реляційної моделі, то другий рівень будується як програмний інтерфейс, що пов'язує клієнтські компоненти з прикладною логікою бази даних.

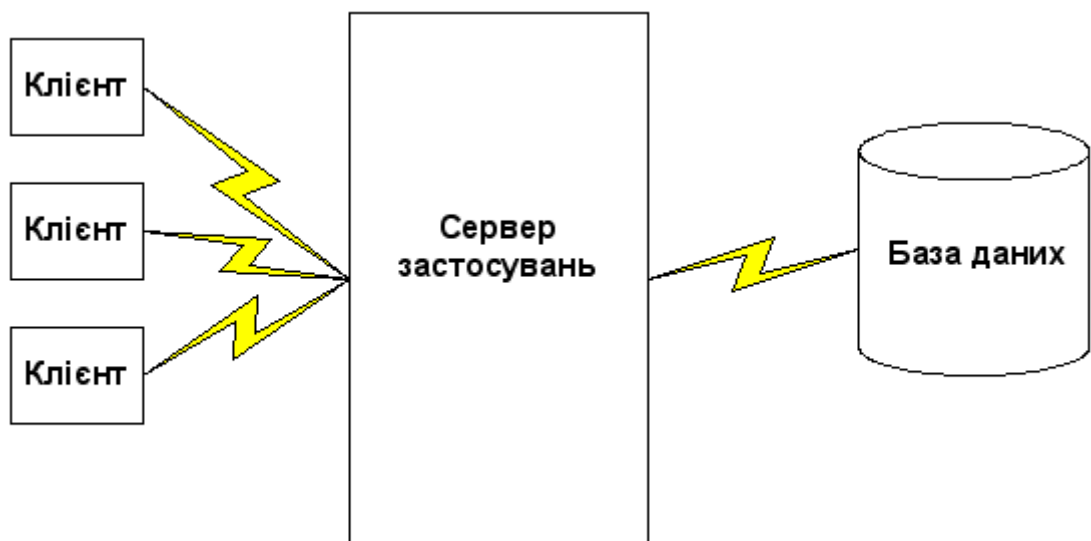


Рисунок 6 – Трирівнева архітектура «клієнт-сервер»

Завдання багаторівневої архітектури – ізолювати рівні один від одного. У трирівневій архітектурі відносно легко можна поміняти СУБД – для цього не потрібно переписувати додаток клієнта заново. Також, відносно легко можна створити новий тип клієнта.

Не рекомендується концентрувати виконання всіх логічних компонентів застосувань на другому рівні. Під час роботи деякі фрагменти можуть частково виконуватися як на першому, так і на третьому рівні. До фрагментів, які, так чи інакше, виконуватимуться на першому рівні, можна віднести:

- перевірку коректності вхідних даних;
- засоби проведення нескладних операцій з даними, вже присутніми на клієнтській системі, наприклад, фільтрацію раніше завантаженої інформації.

Основними перевагами трирівневої архітектури в порівнянні з дворівневою є:

- масштабованість;
- конфігуруємість;
- можливість модифікації бізнес-логіки;
- багатократне використання сервісів.

Клієнт відповідає лише за призначений для користувача інтерфейс і, можливо, виконує деяку дуже просту логічну обробку даних, наприклад перевірку коректності введення даних. Обмежений подібним функціональним набором клієнт отримав назву "тонкого". Основна бізнес-логіка додатка тепер знаходиться на власному виділеному рівні, який фізично пов'язаний з клієнтом і сервером бази даних за допомогою локальної або глобальної обчислювальної мережі. При цьому передбачається, що один сервер додатків може обслуговувати багато клієнтів.

Відділення основної бізнес-логіки додатка від функцій бази даних спрощує завдання рівномірного розподілу навантаження.

Як базова архітектура прийнята трирівнева архітектура «клієнт-сервер», де як тонкий клієнт використовується мобільний додаток на персональному смартфоні кінцевого користувача, як сервер додатку використовується Node.js, а сервер баз даних управляється MongoDB.

2.3 Обґрунтування вибору програмних засобів реалізації системи

Для реалізації поставленого завдання були вибрані наступні засоби:

- MongoDB документо-орієнтована система керування базами даних;
- Robomongo для роботи з СКБД;
- Microsoft Visual Studio Code використовується як текстовий редактор;
- Genymotion для емуляції додатка то його тестування;
- Node.js – в ролі сервера.

В даному розділі наводяться основні відомості та коротка характеристика кожного з вибраних засобів.

В ролі системи керування базами даних було вирішено використовувати MongoDB. Відсутність жорсткої схеми бази даних та, в зв'язку з цим, потреби при щонайменшій зміні концепції зберігання даних переробляти цю схему значно полегшують роботу з базами даних MongoDB та подальшим їх масштабуванням. Ця СКБД повністю підходить для даної роботи, оскільки:

- повноцінно підтримує JavaScript;
- швидко працює;
- має кращу масштабованість;
- має простоту у використанні.

MongoDB це документо-орієнтована система керування базами даних з відкритим сирцевим кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів [10]¹⁾.

Код MongoDB написаний на мові C++ і поширюється в рамках ліцензії GPLv3.

¹⁾ [10] MongoDB 1.0 GA первый релиз новой СУБД. URL: <https://www.nixp.ru/news/9901.html> (дата звернення 10.03.2019).

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій. У MongoDB є вбудовані засоби із забезпечення шардінга (розподіл набору даних по серверах на основі певного ключа), комбінуючи який реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу. Розширення кластера або перетворення одного сервера в кластер проводиться без зупинки роботи БД простим додаванням нових машин.

СКБД управляє наборами JSON-подібних документів, що зберігаються в двійковому вигляді в форматі BSON. Зберігання і пошук файлів в MongoDB відбувається завдяки викликам протоколу GridFS. Подібно до інших документо-орієнтованих СКБД (CouchDB тощо), MongoDB не є реляційною СКБД.

Якщо реляційні бази даних зберігають рядки, то MongoDB зберігає документи. На відміну від рядків документи можуть зберігати складну за структурою інформацію. Документ можна уявити як сховище ключів і значень. Ключ являє просту мітку, з яким асоційована певна область даних.

Однак при всіх відмінностях є одна особливість, яка зближує MongoDB і реляційні бази даних. У реляційних СУБД зустрічається таке поняття як первинний ключ. Це поняття описує якийсь стовпець, який має унікальні значення. У MongoDB для кожного документа є унікальний ідентифікатор, який називається `_id`. І якщо явно не вказати його значення, то MongoDB автоматично згенерує для нього значення.

Кожному ключу зіставляється певне значення. Але тут також треба враховувати одну особливість: якщо в реляційних базах є чітко окреслена структура, де є

поля, і якщо якесь поле не має значення, йому (в залежності від налаштувань конкретної бд) можна привласнити значення NULL. У MongoDB все інакше. Якщо якось ключ не порівнювати значення, то цей ключ просто опускається в документі і не вживається.

Якщо в традиційному світі SQL є таблиці, то в світі MongoDB є колекції. І якщо в реляційних БД таблиці зберігають однотипні жорстко структуровані об'єкти, то в колекції можуть містити найрізноманітніші об'єкти, що мають різну структуру і різний набір властивостей.

Система зберігання даних в MongoDB представляє набір реплік. У цьому наборі є основний вузол, а також може бути набір вторинних вузлів. Всі вторинні вузли зберігають цілісність і автоматично оновлюються разом з оновленням головного вузла. І якщо основний вузол з якихось причин виходить з ладу, то один з вторинних вузлів стає головним.

Robomongo – це кросплатформний інструмент з відкритим вихідним кодом. Robomongo допомагає керувати MongoDB базами даних. Використовує той же движок JavaScript, який підтримує інтерпретатор командного рядка MongoDB. Підтримує аутентифікацію і в цілому має приємний графічний інтерес. На вибір користувачеві дається кілька варіантів перегляду вмісту колекцій – у вигляді дерева, таблиць або plain-text JSON-документів.

Robomongo має двохпанельний режим роботи для різних запитів. Так само підтримує вкладки (кожна зі своїм окремим інтерпретатором командної строки), зберігаючи результати останнього запиту кожної окремої сесії, тому завжди можна порівняти відповідь БД до і після якоїсь важливої зміни або працювати над паралельними завданнями одночасно.

Robomongo має просте, але досить функціональний автодоповнення, яке можна використовувати при ручному складанні запитів до БД. Додатково до всього у Robomongo є невеликий набір сніпетів для найбільш частих CRUD-операцій з документами, які зберігають час при необхідності щось швидко змінити у БД.

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

За основу для Visual Studio Code використовуються напрацювання вільного проекту Atom, що розвивається компанією GitHub. Зокрема, Visual Studio Code є надбудовою над Atom Shell, що використовують браузерний рушій Chromium і Node.js [11]¹⁾.

Продукт підтримує розробку для платформ ASP.NET і Node.js, і позиціонується як легковаге рішення, що дозволяє обійтися без повного інтегрованого середовища розробки. Серед підтримуваних мов і технологій: JavaScript, C++, C#, TypeScript, jade, PHP, Python, XML, Batch, F#, DockerFile, Coffee Script, Java, HandleBars, R, Objective-C, PowerShell, Luna, Visual Basic, Markdown, JSON, HTML, CSS, LESS і SASS.

Як і багато редакторів коду, VS Code використовує поширену схему розташування основних елементів - зліва оглядач файлів, праворуч редактор коду. На додаток до цього є специфічні елементи, що забезпечують навігацію і виконання спеціальних команд.

VS Code працює з файлами і папками в яких знаходяться проекти. У найпростішому випадку ви можете відкрити файл на редагування просто виконавши команду `./code index.html`. Більш цікавим випадком є відкриття папки. VS Code сам визначає тип проекту в залежності від вмісту папки. Наприклад, якщо в папці знаходяться файли `package.json`, `project.json`, `tsconfig.json` або файли `.sln` і `.proj` для Visual Studio ASP.NET 5.0 то VS Code включає багато нових функцій які забезпечують IntelliSense, підказки, навігацію по коду, виконання команд і багато іншого. При кожному запуску VS Code буде завантажено останній стан основних елементів.

На рис.7 показаний інтерфейс середовища розробки.

¹⁾ [11] Visual Studio Code URL: <http://www.opennet.ru/opennews/art.shtml?num=42132> (дата звернення 01.04.2019).

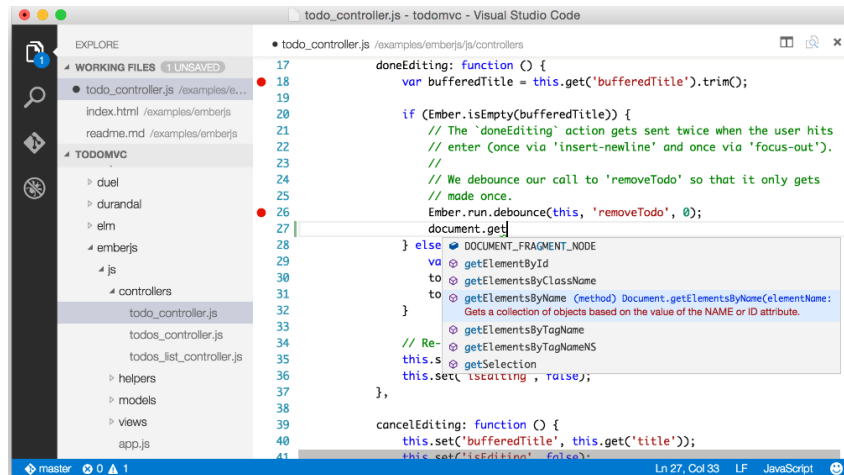


Рисунок 7 – Інтегроване середовище розробки

Редактор VS Code може відкрити для редагування на екрані до трьох файлів одночасно, розташовуючи їх один за одним справа. Найголовнішим інструментом взаємодії з редактором в VS Code є палітра команд. Усі команди, які перераховані у палітрі, теж прив'язані до клавіш.

Genymotion – швидкий емулятор Android, який включає в себе налаштовані образи Android (x86 з апаратним прискоренням OpenGL), ідеальний для тестування програми (рис. 8). Цей проект виріс з старого AndroidVM, і в порівнянні з ним Genymotion має новий дизайн плеєра, установник і багато іншого.

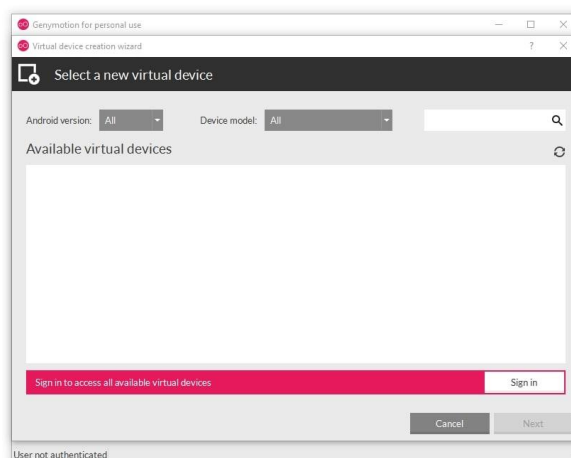


Рисунок 8 – Середовище емулятора Genymotion

Genymotion доступний для Linux, Windows і Mac OS X і вимагає VirtualBox. Код віртуальний машин відкритий, але софт, який працює на хості, безкоштовний для використання, але його вихідний код закритий. В майбутньому Genymotion матиме безкоштовну версію з безліччю можливостей, але також будуть доступні платні версії, в основному для великих компаній, яким потрібна спільна робота над Genymotion.

Можливості Genymotion:

- швидке завантаження і запуск попередньо налаштованих образів Android. Список пристроїв постійно розширюється;
- GPS (з налаштованим координатами) і батарея (з налаштованим рівнем батареї);
- відображення: апаратне прискорення з OpenGL, режим повноекранного перегляду;
- консоль, яка дозволяє вам взаємодіяти з вашою віртуальною машиною, використовуючи командний рядок;
- емуляція передньої і задньої камери;
- модулі до Eclipse і Android Studio;
- підтримує Linux, Windows і Mac;
- загальний буфер обміну віртуального пристрою і ПК.

Щоб запустити програму, досить зареєструвати аккаунт (логін і пароль надалі будуть потрібні для входу в емулятор) на офіційному сайті і отримати посилання на інсталятор. Після установки можна обрати і завантажити зі списку віртуальні пристрої, чиї параметри відповідають реальним смартфонам. Принцип роботи тут схожий з Android SDK: спочатку додається пристрій в менеджері, потім необхідно його налаштувати і запустити. Як засіб управління можна використовувати і звичайну, і віртуальну клавіатуру.

JavaScript (JS) – динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що викону-

ється на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. Мова JavaScript також використовується для програмування на стороні сервера (подібно до таких мов програмування, як Java і C#), розробки ігор, стаціонарних та мобільних додатків, сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF-документів тощо.

JavaScript може виконуватися не тільки в браузері, а де завгодно, потрібна лише спеціальна програма – інтерпретатор. Процес виконання скрипта називають «інтерпретацією». До всіх основних браузерів вбудовано інтерпретатор JavaScript, саме тому вони можуть виконувати скрипти на сторінці. Але, зрозуміло, JavaScript можна використовувати не тільки в браузері. Це повноцінна мова, програми на якому можна запускати та на сервері, та навіть в пральній машинці, якщо в ній встановлений відповідний інтерпретатор.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme [12]¹⁾.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою C має такі корінні відмінності:

¹⁾ [12] JavaScript. URL: <https://uk.wikipedia.org/wiki/JavaScript> (дата звернення 20.04.2019).

- об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів;
- функції як об'єкти першого класу;
- обробка винятків;
- автоматичне приведення типів;
- автоматичне прибирання сміття;
- анонімні функції.

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

JavaScript спочатку створювався для того, щоб зробити web-сторінки «живими». Програми на цій мові називаються скриптами. У браузері вони підключаються безпосередньо до HTML та, як тільки завантажується сторінка, тут же виконуються. Мова JavaScript унікальна завдяки своїй повній інтеграції з HTML / CSS.

React Native – це фреймворк для розробки кроссплатформенних додатків для iOS і Андроїд. Додаток будується з компонент платформи – це нативні модулі, вбудовані в React-компоненти.

Стилі в React Native пишуться за принципом Flexbox, що застосовується в CSS. Flexbox – підхід до компонування div-елементів. Раніше писали "float: left", "float: right", "margin-right" тощо. Зараз Flexbox – це нова, більш гнучка модель. Саме вона використовується в React Native для написання стилів. Але тут вони пишуться не в CSS, а в JS – потім вони переводяться на мову, зрозумілу платформі. Припустимо, "margin-top" перекладається в зрозумілий для iOS відступ зверху, або в відступ зверху, зрозумілий для Android.

Node.js – платформа з відкритим кодом для виконання високопродуктивних мережеских застосувань, написаних мовою JavaScript. Платформа node.js перетворила мову JavaScript, що в основному використовувалась в браузерах на мову загального використання з великою спільнотою розробників.

Node.js характеризується такими властивостями:

- асинхронна однопотокова модель виконання запитів;
- неблокуючий ввід/вивід;
- система модулів CommonJS.

Для керування модулями використовується пакетний менеджер npm (node package manager).

Node.js призначений для відокремленого виконання високопродуктивних мережеских застосунків на мові JavaScript. Функції платформи не обмежені створенням серверних скриптів для веб, платформа може використовуватися і для створення звичайних клієнтських і серверних мережеских програм. Для забезпечення виконання JavaScript-коду використовується розроблений компанією Google рушій V8.

Для забезпечення обробки великої кількості паралельних запитів у Node.js використовується асинхронна модель запуску коду, заснована на обробці подій в неблокуючому режимі та визначенні обробників зворотніх викликів (callback). Як способи мультиплексування з'єднань підтримується epoll, kqueue, /dev/poll і select. Для мультиплексування з'єднань використовується бібліотека libuv, для створення пулу потоків (thread pool) задіяна бібліотека libeio, для виконання DNS-запитів у неблокуючому режимі інтегрований c-ares. Всі системні виклики, що спричиняють блокування, виконуються всередині пула потоків і потім, як і обробники сигналів, передають результат своєї роботи назад через неіменовані канали (pipe).

У проєкті використовується фреймворк Express. Express – це мінімалістичний і гнучкий веб-фреймворк для додатків Node.js, що надає великий набір функцій для мобільних і веб-додатків.

Express.js, або просто Express – програмний каркас розробки веб-застосунків для Node.js, реалізований як вільне і відкрите програмне забезпечення під ліцензією MIT. Він спроектований для створення веб-застосунків і API. Де-факто є стандартним каркасом для Node.js. Автор фреймворка, TJ Holowaychuk, описує його як створений на основі написаного на мові Ruby каркаса Sinatra, маючи на увазі, що він мінімалістичний, але має велику кількість плагінів, що підключаються.

2.4 Проектування інформаційної системи за допомогою методології функціонального моделювання

При проектуванні мобільного застосування для визначення мінералів була обрана методологія SADT.

SADT (акронім від англ. Structured Analysis and Design Technique) – це методологія структурного аналізу та проектування, інтегруюча процес моделювання, управління конфігурацією проекту, використання додаткових мовних засобів і керівництво проектом зі своєю графічною мовою. Процес моделювання може бути розділений на декілька етапів: опитування експертів, створення діаграм і моделей, поширення документації, оцінка адекватності моделей і прийняття їх для подальшого використання. Цей процес добре налагоджений, тому що при розробці проекту фахівці виконують конкретні обов'язки, забезпечуючи своєчасний обмін інформацією.

За допомогою графічної мови IDEF0, інформаційно-пошукова система постає у вигляді набору взаємопов'язаних функціональних блоків. Моделювання засобами IDEF0, як правило, є першим етапом вивчення системи.

IDEF0-моделі складаються з трьох типів документів: графічних діаграм, тексту і глосарію. Ці документи мають перехресні посилання один на одного. Графічна діаграма – головний компонент IDEF0-моделі, що містить блоки, стрілки, з'єднання блоків і стрілок та асоційовані з ними відносини. Блоки представляють основні функції об'єкта що моделюється. Ці функції можуть

бути розбиті (декомпозиція) на складові частини та представлені у вигляді більш докладних діаграм; процес декомпозиції продовжується до тих пір, поки об'єкт не буде описаний на рівні деталізації, необхідному для досягнення цілей конкретного проекту. Діаграми верхнього рівня забезпечує найбільш загальне або абстрактне описання об'єкта моделювання. За цією діаграмою слідує серія дочірніх діаграм, що дають більш детальне уявлення про об'єкт.

Кожна модель повинна мати контекстну діаграму верхнього рівня, на якій об'єкт моделювання представлений єдиним блоком з граничними стрілками. Ця діаграма називається А-0. Стрілки на цій діаграмі відображають зв'язок об'єкта моделювання з навколишнім середовищем. Оскільки єдиний блок представляє весь об'єкт, його ім'я загальне для всього проекту. Це ж справедливо і для всіх стрілок діаграми, оскільки вони представляють повний комплект зовнішніх інтерфейсів об'єктивним та. Діаграма А-0 встановлює область моделювання та її межу.

Під CASE-системою розуміють програмні засоби, що підтримують процеси створення і супроводження інформаційних систем, включно з аналізом і формулюванням вимог, проектуванням прикладного програмного забезпечення і баз даних, генеруванням коду, тестуванням, документуванням, забезпеченням якості, конфігураційним керуванням і керуванням проектом, а також іншими процесами. CASE-засоби разом із системним програмним забезпеченням і технічними засобами утворюють повне середовище розробки інформаційної системи.

Програмне забезпечення Ramus призначене для використання в проектах, в яких необхідний опис бізнес-процесів підприємства і (або) створення систем класифікації і кодування. Ramus повністю підтримує методологію моделювання бізнес-процесів IDEF0 і DFD, а так само має ряд додаткових можливостей покликаних задовольнити потреби команд розробників систем управління підприємствами. Ramus дозволяє створювати графічні моделі бізнес-процесів згідно правил синтаксису IDEF0, використовуючи власний графічний редактор. Крім того, Ramus дозволяє створювати систему класи-

фікації і кодування всіх об'єктів, які фігурують в бізнес-процесах підприємства і зв'язати цю систему з графічними моделями бізнес-процесів. Ramus володіє гнучкими можливостями побудови звітності після графічних моделей бізнес-процесів і системи класифікації і кодування. Дана можливість дозволяє створювати звітність у формі документів, які регламентують діяльність підприємства. Наприклад, регламенти процесів і посадові інструкції. Причому, дана регламентуюча документація, будучи такою, що автоматично генерується з моделей процесів і системи класифікації і кодування, носить системний і несуперечливий характер, що критично важливе при побудові систем управління підприємствами. Важливо, що Ramus дозволяє переглядати вміст проєктів через веб-сервер-інтерфейс. Це спрощує комунікації між розробниками і користувачами систем управління, створеними з використанням можливостей Ramus.

Контекстна діаграма інформаційної системи представлена на рис. 9.

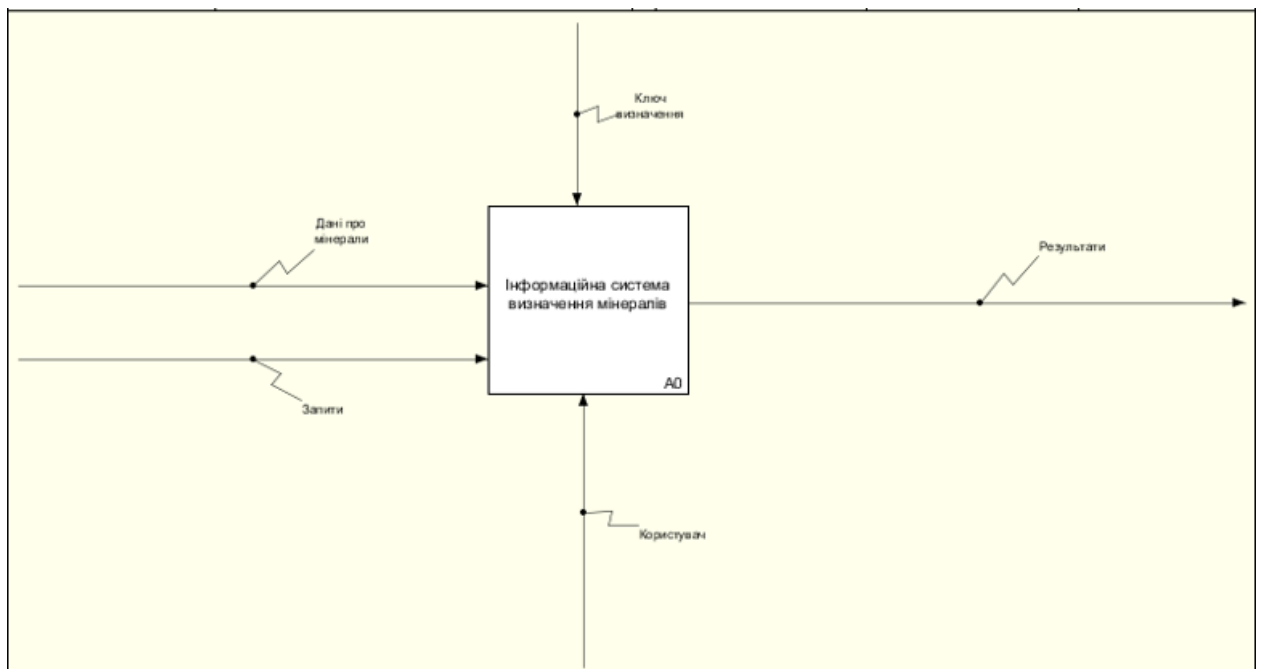


Рисунок 9 – Контекстна діаграма

Після декомпозиції контекстної діаграми проводиться декомпозиція кожного великого фрагмента системи на більш дрібні і так далі, до досягнення потрібного рівня подробности опису. Після кожного сеансу декомпозиції проводяться сеанси експертизи – експерти предметної області вказують на відповідність реальних процесів створеним діаграмам. Знайдені невідповідності виправляються, і тільки після проходження експертизи без зауважень можна приступати до наступного сеансу декомпозиції. Так досягається відповідність моделі реальним процесам на кожному рівні декомпозиції моделі. Синтаксис опису системи в цілому і кожного її фрагмента однаковий у всій моделі.

Після декомпозиції контекстної діаграми отримуємо 3 блоки – роботи. Ці блоки представляють основні підфункції початкової функції. Діаграма декомпозиції наведена на рис. 10.

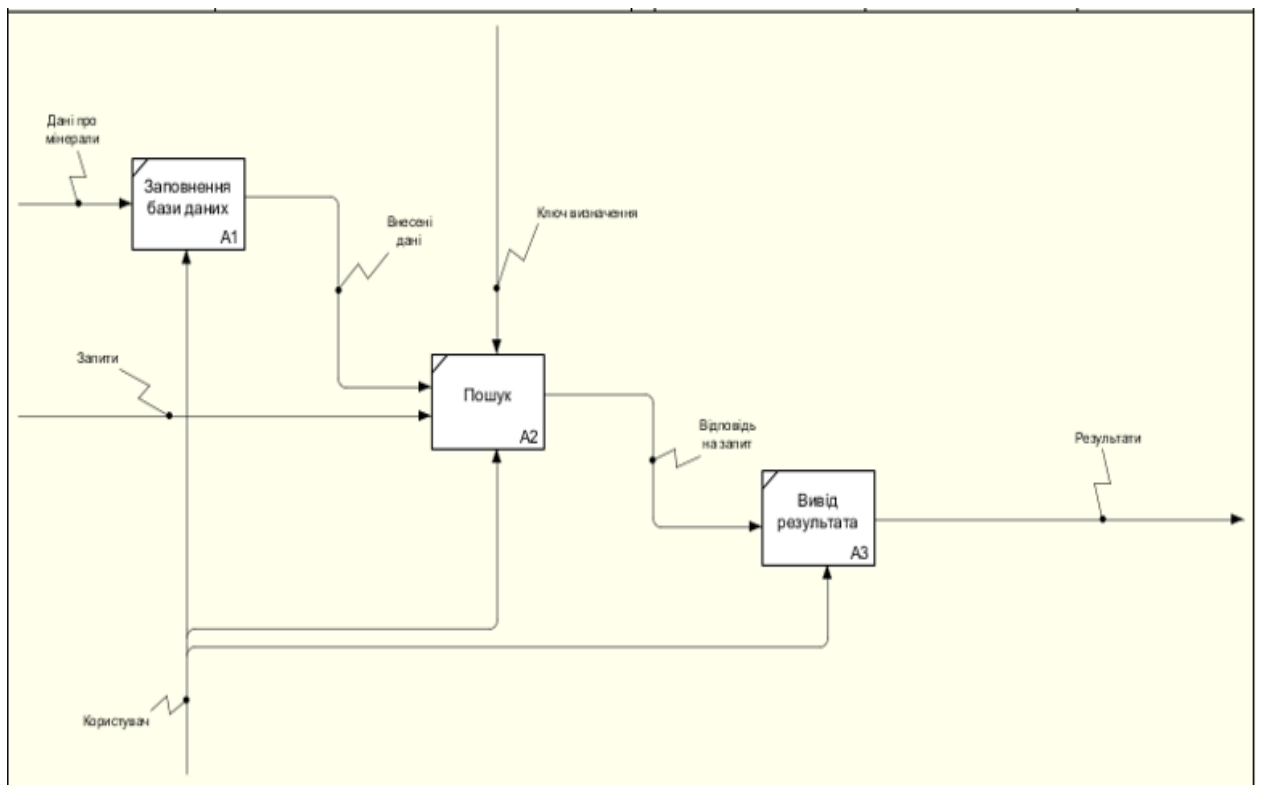


Рисунок 10 – Діаграма декомпозиції

Майже всі CASE-засоби структурного моделювання підтримують можливість побудови DFD-діаграм. Зовнішній вигляд компонентів може відрізнятися, але набір залишається незмінним. Основні компоненти DFD роботи (процеси), зовнішні сутності, потоки даних, накопичувачі даних (сховища). Джерела інформації (зовнішня сутність) породжують інформаційні потоки (потоки даних), що переносять інформацію до підсистем або процесів. Ті в свою чергу перетворюють інформацію і породжують нові потоки, які переносять інформацію до інших процесів або підсистем, накопичувачів даних або зовнішньої сутності – споживачам інформації.

У DFD роботи (процеси) представляють собою функції системи, що перетворюють входи у виходи. Хоча роботи зображуються прямокутниками з закругленими кутами (або овалами), сенс їх збігається зі змістом робіт IDEF0. Так само, як процеси IDEF0, вони мають входи і виходи, але не підтримують управління та механізми.

Процеси у програмі Ramus Educational зображають овалом.

Зовнішні сутності зображують входи в систему і/або виходи з системи. Зовнішні сутності зображуються у вигляді прямокутника з тінню і зазвичай розташовуються по краях. Одна зовнішня сутність може бути використана багаторазово на одній або декількох діаграмах. Зазвичай такий прийом використовують, щоб не малювати занадто довгих і заплутаних стрілок.

Потоки зображуються стрілками і описують рух об'єктів з однієї частини системи в іншу. Оскільки в DFD кожна сторона процесу не має чіткого призначення, як в IDEF0, стрілки можуть підходити і виходити з будь-якої грані процесу. У системах обробки інформації сховища даних є механізмом, який дозволяє зберегти дані для подальших процесів.

На рис. 11 представлена контекстна діаграма потоків даних, що описує головний процес майбутньої системи – організація роботи інформаційної системи.

Зовнішньою сутністю є клієнт – відвідувач системи. Це фізична особа, яка є джерелом або приймачем інформації. Визначення деякого об'єкту або системи як зовнішньої сутності вказує на те, що вона знаходиться за межами меж аналізованої інформаційної системи. В процесі аналізу деяка зовнішня сутність може бути перенесена всередину діаграми аналізованої системи, якщо це необхідно, або, навпаки, частина процесів може бути винесена за межі діаграми і представлена як зовнішня сутність.

Потік даних визначає інформацію, яка передається через деяке з'єднання від джерела до приймача.

Накопичувач даних є абстрактним пристроєм для зберігання інформації, яку можна у будь-який момент помістити в накопичувач і через деякий час витягнути, причому способи приміщення і витягання можуть бути будь-якими.

Основними накопичувачами даних в роботі є Характеристики та Мінерали.

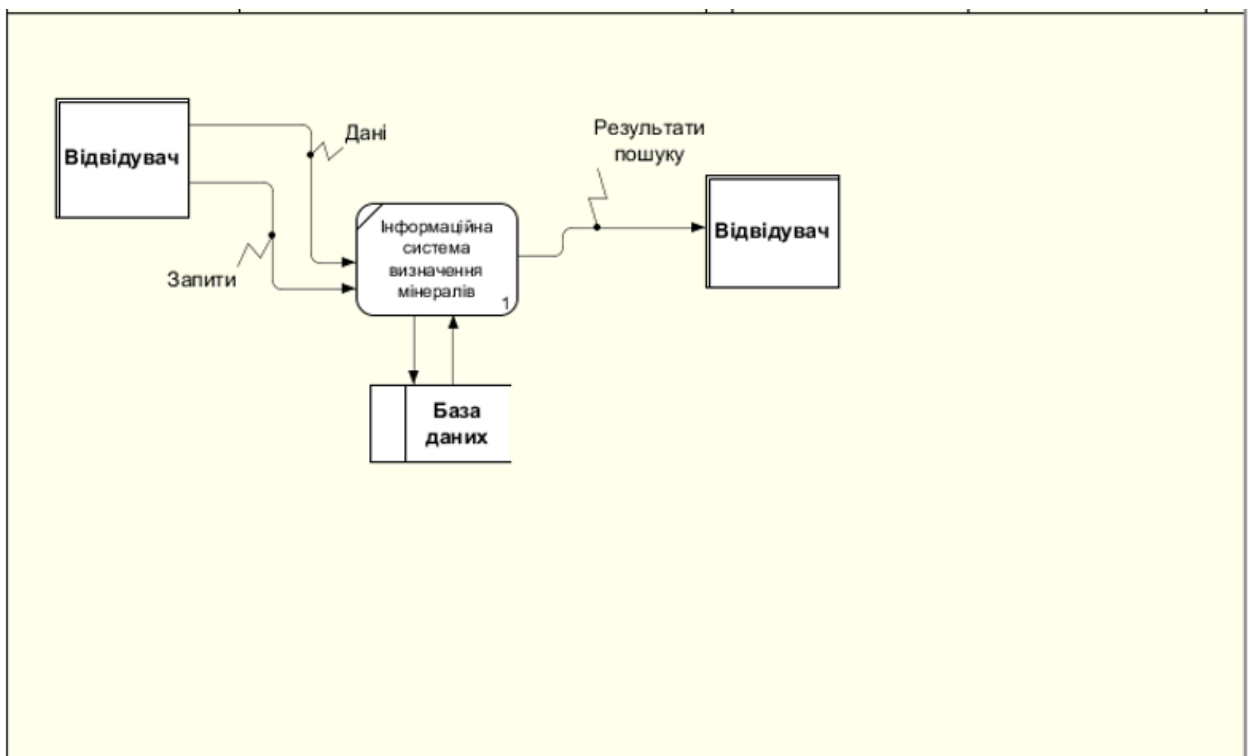


Рисунок 11 – Контекстна діаграма потоків даних

Діаграми потоків даних також піддаються декомпозиції. Модель найвищого рівня називається контекстною діаграмою. Діаграми верхніх рівнів ієрархії визначають основні процеси системи із зовнішніми входами і виходами. Вони деталізуються за допомогою діаграм нижнього рівня.

У ході деталізації були виявлені наступні процеси системи:

- перегляд мінералів з каталогу;
- пошук інформації про необхідний вид мінерала;
- надання інформації;
- внесення та корегування даних.

На рис. 12 показана деталізована діаграма потоків даних.

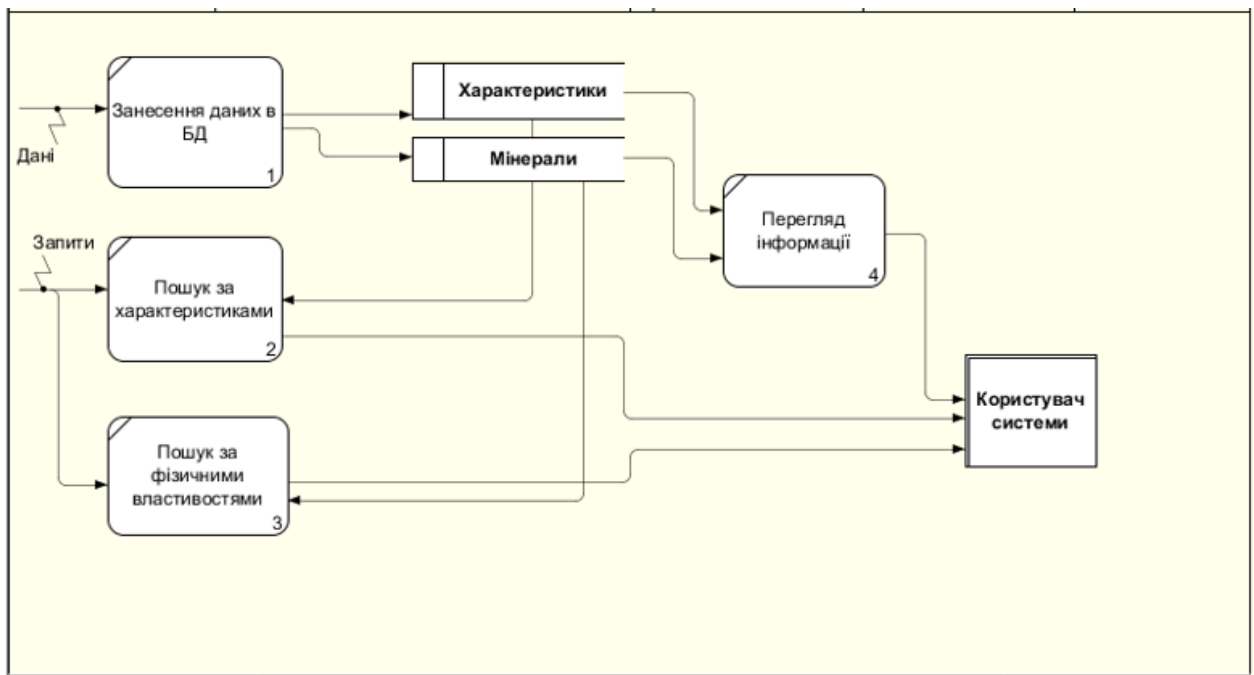


Рисунок 12 – Деталізована діаграма потоків даних

Після побудови закінченої моделі системи її необхідно верифікувати, тобто перевірити на повноту і узгодженість. У повній моделі всі її об'єкти (підсистеми, процеси, потоки даних) повинні бути детально описані і деталізовані. Виявлені недеталізовані об'єкти слід деталізувати, повернувшись на по-

передні кроки розробки. У узгодженій моделі для всіх потоків даних і накопичувачів даних повинне виконуватися правило збереження інформації: всі дані, що поступають куди-небудь, повинні бути прочитані, а всі зчитувані дані повинні бути записані.

2.5 Специфікація вимог до бази даних

Першим етапом при проектуванні баз даних є аналіз обраної області. Завдання цього етапу полягають, по-перше, в здобутті проектувальником загального уявлення про обрану область; по-друге, в зборі елементів даних, які потрібно відображати в базі даних; по-третє, у формулюванні обмежень цілісності, яким повинні задовольняти дані майбутньої бази.

Результатом виконання цієї фази проекту з'явилася підготовка специфікацій вимог, характерних для даної системи. У специфікації зафіксовані вимоги до інформації, яка буде поміщена в створювану базу даних.

Головні характеристики, за якими зручно здійснювати пошук мінералу, розбиті на дві групи:

- зовнішній вигляд;
- фізичні властивості.

У розділі «Зовнішній вигляд» можна обирати такі характеристики:

- колір;
- колір риси;
- прозорість;
- блиск.

Та у розділі «Фізичні властивості» можна обирати наступні характеристики:

- злам;
- твердість;
- спайність;
- сингонія;

– класифікація.

Для кожного мінералу слід мати його зображення, опис всіх характеристик і додаткову інформацію.

Визначення мінералів включає завдання декількох параметрів і відбір групи мінералів, відповідних даному набору.

За аналізом обраної області був спроектований ключ визначення мінералів за характеристиками (рис.13).



Рисунок 13 – Дихотомічний ключ

2.6 Побудова моделі даних

У даному проєкті використано документно-орієнтовану СКБД.

Документно-орієнтована система керування базами даних – це система керування базами даних, спеціально призначена для зберігання ієрархічних структур даних (документів) і зазвичай реалізована за допомогою підходу NoSQL. В основі документно-орієнтованих СКБД лежать документні сховища,

котрі мають структуру дерева (іноді лісу). Структура дерева починається з кореневого вузла і може містити кілька внутрішніх і листових вузлів. Листові вузли містять дані, які при додаванні документа заносяться в індекси, що дозволяє навіть при досить складній структурі знаходити місце (шлях) шуканих даних. АРІ для пошуку дозволяє знаходити за запитом документи та частини документів. На відміну від сховищ типу ключ-значення, вибірка за запитом до документного сховища може містити частини великої кількості документів без повного завантаження цих документів в оперативну пам'ять. Документи можуть бути організовані (згруповані) в колекції. Їх можна вважати віддаленим аналогом таблиць реляційних СКБД, але колекції можуть містити інші колекції. Хоча документи колекції можуть бути довільними, для ефективнішого індексування краще об'єднувати у колекцію документи зі схожою структурою [13]¹⁾.

Документо-орієнтована СКБД звертається до документів через унікальний ключ, який представляє документ. Цей ключ часто є простим рядком, URI або шляхом. Ключ використовується для отримання документу з бази даних. Звичайно СКДБ зберігає індекс ключів для прискорення одержання документів. Центральним поняття документо-орієнтованих баз даних є Документ. Деталі визначення цієї головної концепції різняться у різних реалізаціях документо-орієнтованих баз даних, но загалом всі смороду мають на увазі, що документ ізолює дані та кодує їх (або інформацію) у певного формату або кодуванні. Це кодування може використовувати XML, YAML, JSON, BSON, а також бінарні формати, такі як PDF чи документи Microsoft Office (MS Word, Excel, тощо).

У даній роботі створено два документа: `characteristics.json` та `minerals.json`. Файл (`characteristics.json`) містить у собі усі характеристики мінералів (рис. 14).

¹⁾ [13] Документо-орієнтована система керування базами даних. URL: https://uk.wikipedia.org/wiki/Document-oriented_database (дата звернення 18.04.2019).

Key	Value	Type
<ul style="list-style-type: none"> (1) ObjectId("592ee1202a0aa633342fe798") <ul style="list-style-type: none"> _id title properties <ul style="list-style-type: none"> [0] <ul style="list-style-type: none"> title collapsed types [1] <ul style="list-style-type: none"> title collapsed types [2] <ul style="list-style-type: none"> title collapsed types [3] <ul style="list-style-type: none"> title collapsed types 	<pre>{ 3 fields } ObjectId("592ee1202a0aa633342fe798") Внешний вид [4 elements] { 3 fields } Цвет true [16 elements] { 3 fields } Цвет черты true [11 elements] { 3 fields } Прозрачность true [3 elements] { 3 fields } Блеск true [8 elements]</pre>	<pre>Object ObjectId String Array Object String Boolean Array Object String Boolean Array Object String Boolean Array Object String Boolean Array Object</pre>
<ul style="list-style-type: none"> (2) ObjectId("592ee1202a0aa633342fe799") <ul style="list-style-type: none"> _id title properties <ul style="list-style-type: none"> [0] [1] [2] 	<pre>{ 3 fields } ObjectId("592ee1202a0aa633342fe799") Физические свойства [5 elements] { 3 fields } { 3 fields } { 3 fields }</pre>	<pre>Object ObjectId String Array Object Object Object</pre>

Рисунок 14 – Відображення характеристик мінералів в Robomongo

Другий файл (minerals.json) містить інформацію про мінерали, тобто: назву, опис, характеристики, зображення (рис. 15).

Key	Value	Type
<ul style="list-style-type: none"> (1) ObjectId("592ee1202a0aa633342fe760") <ul style="list-style-type: none"> _id name description properties <ul style="list-style-type: none"> [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] img (2) ObjectId("592ee1202a0aa633342fe761") (3) ObjectId("592ee1202a0aa633342fe762") (4) ObjectId("592ee1202a0aa633342fe763") (5) ObjectId("592ee1202a0aa633342fe764") (6) ObjectId("592ee1202a0aa633342fe765") (7) ObjectId("592ee1202a0aa633342fe766") (8) ObjectId("592ee1202a0aa633342fe767") (9) ObjectId("592ee1202a0aa633342fe768") (10) ObjectId("592ee1202a0aa633342fe769") 	<pre>{ 5 fields } ObjectId("592ee1202a0aa633342fe760") Эвдиалит Относительно редкий акцессорный минерал щелочных горн... [12 elements] silicates hardness5 colorViolet colorRed colorYellow colorTraitsWhite shineGlass transparent turbid cleavagelmpfect fractureUneven trigonal evdialid.jpg { 5 fields } { 5 fields } { 5 fields } { 5 fields } { 5 fields } { 5 fields } { 5 fields } { 5 fields } { 5 fields }</pre>	<pre>Object ObjectId String String Array String String String String String String String String String String String String String String String String Object Object Object Object Object Object Object Object Object Object</pre>

Рисунок 15 – Відображення набору мінералів в Robomongo

Схема даної БД виглядає наступним чином (рис.16):



Рисунок 16 – Модель даних БД

2.7 Побудова фізичної моделі даних

Процес побудови фізичної моделі даних виконано за допомогою СКБД MongoDB. Також обрано найбільш відповідну для вирішення нашого завдання файлову структуру для СУБД Robomongo.

Інтерфейс мобільного додатка написаний за допомогою мови програмування JavaScript з використанням фреймворку React Native. Кожен компонент додатка описаний в різних файлах, що дозволяє використовувати їх неодноразово. Також можна використовувати компонент в компоненті, що робить наш додаток досить зручним для розробки і подальшого оновлення.

Сервер написаний на програмній платформі Node.js з використанням фреймворку Express. Ініціалізується в файлі Minerals-server/index.js. Також в цьому файлі прописано заповнення БД. При запуску сервера перевіряється чи заповнено БД даними. Якщо дані відсутні, то вони імпортуються з JSON-файлів.

3 РЕАЛІЗАЦІЯ СИСТЕМИ

Метою даної роботи є розробка інформаційної системи та застосування для класифікації та діагностики мінералів за допомогою мобільних додатків.

Робота з системою починається з встановлення мобільного додатка на смартфон. Це може бути будь-який смартфон, тому що ця програма є кроссплатформенною. Працювати з системою може будь-який користувач. Реєстрація для використання програми не потрібна. Користувач може вільно шукати та переглядати інформацію про цікаві для його мінерали.

Спочатку на комп'ютер необхідно встановити програму Node.js. На сайті React Native [14]¹⁾ описується як встановити цю програму. Android Studio встановлювати необов'язково. Таким же чином встановлюємо програми Genymotion та Robomongo. Далі діємо за планом: запускаємо Genymotion, запускаємо Robomongo, відкриваємо сервер Minerals-server через командний рядок за допомогою команди `npm start`. Важливо перевірити порт сервера, він повинен співпадати з портом, який вказаний в `Api.js`.

Заходимо у теку Minerals та в командному рядку вводимо команду `npm android`. На емуляторі запуситься застосування, в якому можна виконувати необхідну роботу.

3.1 Структура та компоненти визначника

Додаток включає функції визначення і пошуку мінералів. Для зручності користувачів передбачається розділення властивостей мінералів на зовнішні та фізичні. Під час пошуку необхідного мінералу, користувач обирає параметри, такі як: колір, фактура, твердість та інші. Далі програма визначає мінерал за обраними параметрами та користувач бачить на екрані смартфона результат, який сгенерувала програма.

¹⁾ [14] Getting Started React Native. URL: <http://reactnatedocs.ru/docs/getting-started.html> (дата звернення 24.05.2019).

Визначення мінералів включає завдання декількох позицій ключа. Якщо за заданими характеристиками мінерал не буде знайдений, то користувачеві будуть представлені можливі варіанти мінералів, які збігаються більш ніж на два параметра, обраними користувачем.

Перелічимо основні програмні компоненти визначника і коротко опишемо виконувані ними функції:

- index.js – файл який ініціалізує додаток;
- Routes/index.js – файл в якому оголошуються сторінки додатка;
- Components/Collapsed.js – файл в якому описаний компонент властивостей мінералів;
- Components/CollapsibleView.js – файл в якому описаний компонент різних параметрів для властивостей мінералів;
- Components/Filters.js – дана сторінка містить параметри, за якими здійснюється пошук мінералів;
- Components/Find.js – на цій сторінці відображається результат пошуку мінералів;
- Components/Home.js – це домашня сторінка, яка відповідає за пошук мінералів;
- Components/Mineral.js – файл в якому оголошується інформація про мінерали;
- Components/SwitchView.js – файл, в якому описаний компонент switch.
- Config/Api.js – файл, в якому знаходиться адресу порту сервера.

Розширення JS – це текстовий файл, що містить код JavaScript, який використовується під час завантаження веб-сторінки. Прикладами JS скриптів можуть бути функції, які відкривають та закривають вікна, перевірка полів форми, що дозволяє змінювати зображення, або створювати меню, що випадає. На файли JS посилаються HTML файли всередині секції HEAD, які використовують функції JavaScript.

3.2 Організація інтерфейсу

Розробка інтерфейсу мобільного додатка – найважливіший етап створення програмного продукту. Від даної роботи залежить те, як користувач сприйме додаток, чи сподобається воно йому, чи буде додаток зручним в експлуатації, та чи стане воно популярним.

Робота системи будується таким чином, що кожна сторінка виконує окрему функцію. Це спрощує роботу з програмою. Вона стає більш зрозумілою та доступною. Також це зручно при розробці програми, бо вона представляє собою конструктор з різних елементів, що значно економить час при реалізації та робить програму більш компактною, тобто вона не займає багато місця в пам'яті телефону.

На головній сторінці користувач бачить назву програми та, за бажанням, може натиснути на кнопку "Поиск", якщо хоче визначити якийсь мінерал.

На сторінці пошуку користувач має змогу обрати різні характеристики мінералів. Коли користувач визначається з вибором необхідних властивостей, він натискає кнопку "Найти" та переходить на наступну сторінку з результатами пошуку.

На сторінці результату пошуку можна побачити всі мінерали, які відповідають необхідним характеристикам. Якщо співпадання з обраними параметрами відсутнє, то користувач побачить відповідний текст. Так само на цій сторінці є кнопка "Назад", яка повертає користувача на сторінку пошуку. При цьому всі вибрані параметри залишаються.

3.3 Дизайн системи

Мобільний дизайн є галуззю дизайну, яка почала розвиватися декілька років тому. З розвитком все більшої кількості мобільних додатків, дизайн таких додатків почав ставати більш важливим, щоб зробити зручне застосування для користувачів.

Основними вимогами до дизайну системи є її простота та єдиний стиль. З цією метою розроблена власна система стилів. Зокрема визначені стилі для заголовків, для кнопок, фільтрів, елементів відображення мінералів.

У роботі використовується стиль мінімалізму. Цей стиль в дизайні, що характеризується лаконічністю, простотою, точністю і ясністю композиції. Джерела мінімалізму лежать в конструктивізмі і функціоналізмі. Кожна сторінка має свій власний простий дизайн.

3.4 Опис роботи з програмою

Головна сторінка інформаційної системи визначника мінералів наведена на рис. 17. Це перша сторінка додатка яку бачить користувач. Для продовження роботи з програмою користувач повинен натиснути на кнопку "Поиск".

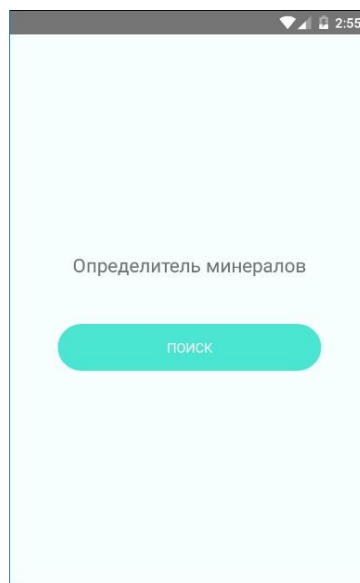


Рисунок 17 – Головна сторінка визначника

При натисканні кнопки "Поиск" користувач переходить на наступну сторінку (рис. 18). На цій сторінці він бачить за якими основними параметрами можна починати пошук мінералу. Характеристики мінералів поділяються на два типи: за зовнішнім виглядом та за фізичними властивостями. Залежно від

потреб користувач обирає за якими характеристиками йому необхідно визначити мінерал.

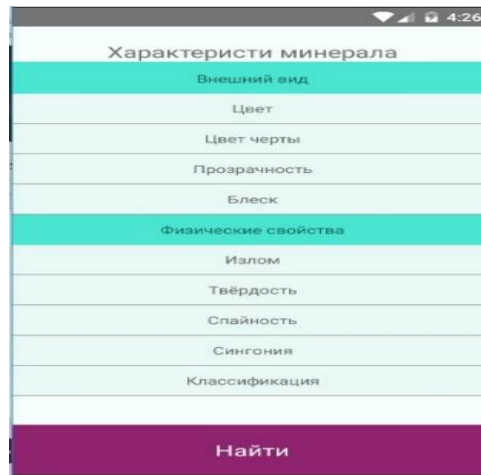


Рисунок 18 – Сторінка пошуку

При натисканні на будь-який параметр, який потрібен користувачу, відкриваються фільтри. Ці фільтри можна обрати натиснувши на перемикач, який розташований перед назвою фільтра. Під фільтрами маєтсья на увазі будь-яка характеристика. За обраними характеристиками буде здійснюватися пошук необхідного мінералу (рис. 19).

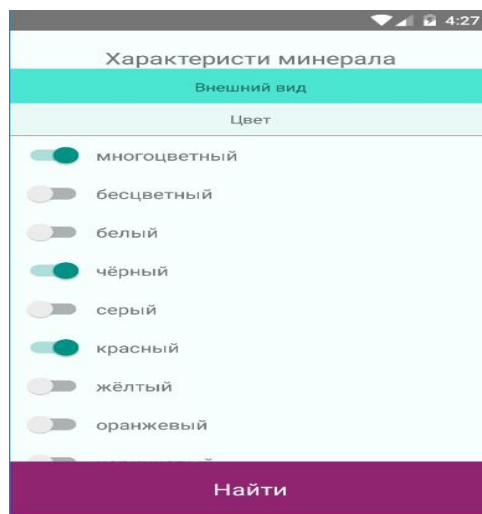


Рисунок 19 – Сторінка пошуку с обраним параметром

Якщо користувач хоче подивитися список всіх мінералів, які є в базі даних, то він може натиснути на кнопку «Поиск».

Додаток перейде на сторінку пошуку, де можна переглянути список всіх мінералів (рис. 20).

Це створено для того, щоб користувач мав повний доступ до інформації щодо мінералів, які зберігаються в нашій базі даних. Тобто ця функція слугуватиме користувачеві довідником мінералів.



Рисунок 20 – Сторінка зі списком всіх мінералів

Якщо користувачеві слід визначити мінерал за певними критеріями, він обирає необхідні характеристики мінералу.

Коли користувач обере потрібні параметри, він може натиснути на кнопку «Найти». Додаток перейде на сторінку результату. Якщо такий мінерал існує, то користувач побачить опис мінералу та його зображення (рис. 21).

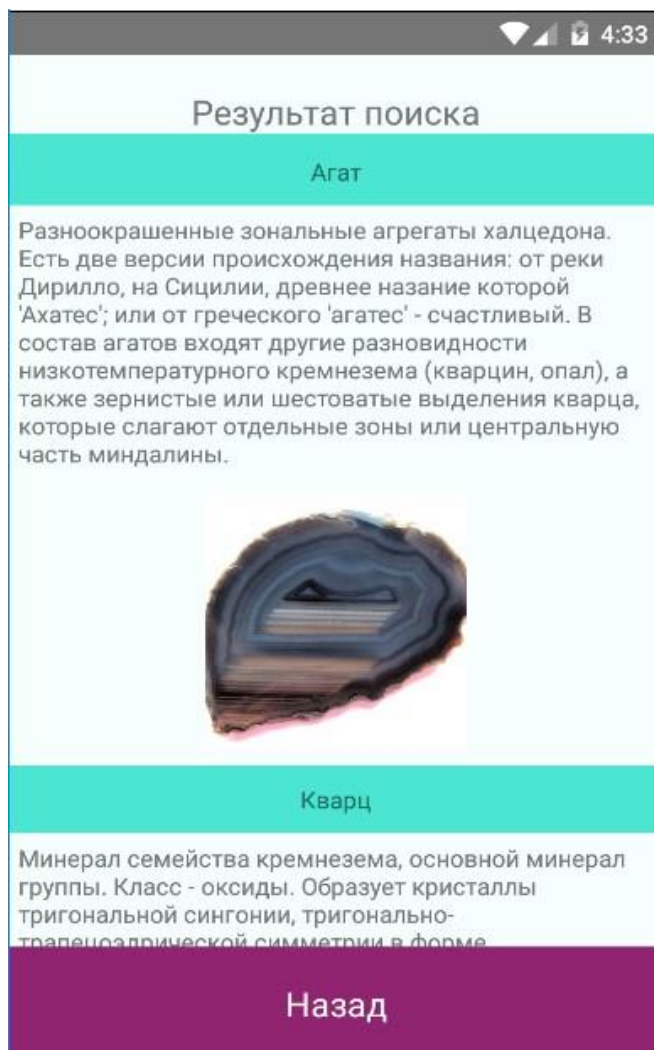


Рисунок 21 – Сторінка результату пошуку з описом мінералів

Якщо такого мінералу не знайшлося, програма про це повідомить (рис. 22). Це може означати, що або такого мінералу не існує в природі, або його ще немає в нашій базі даних. В такому випадку можна змінити, видалити деякі вибрані параметри. Або переглянути весь список мінералів і самостійно визначити мінерал по фото.



Рисунок 22 – Сторінка результату пошуку

Якщо користувач захоче повернутися назад, для пошуку нового мінералу або зміни параметрів, він може натиснути на відповідну кнопку, яка поверне його на сторінку пошуку.

ВИСНОВКИ

В ході виконання бакалаврської кваліфікаційної роботи роботи була розроблена інформаційна система, яка дозволяє визначити необхідний мінерал.

Під час виконання даної роботи була організована та описана база даних та специфікація вимог до бази даних. Також були обрані найбільш актуальні для даної теми сучасні технології, які дозволили досягнути головної мети даного проекту.

Головним завданням було сформувати базу даних таким чином, щоб вона мала універсальний характер тобто, щоб була можливість зберігати данні про будь-які мінерали, а також ефективно маніпулювати цими даними.

Також був реалізований інтерфейс мобільного застосування з використанням сучасних технологій. Основним завданням в реалізації програми було зробити її максимально зручною та простою для користувачів.

Мета, яку мала робота була досягнута та реалізована в повному обсязі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Библиотечно-библиографическая классификация для организации библиотечных фондов. URL:<http://www.elbib.ru/index.phtml?page=elbib/rus/SI> (дата звернення 24.02.2019).
2. Павлишин В. Вступ до мінералогії (курс лекцій). К.:Вид-во ДГЦУ, 1997. 80 с.
3. Поваренных А. С. Кристаллохимическая классификация минеральных видов. К.: Наук. думка, 1966.132 с.
4. Балковский Б. Е. Цифровой политомиический ключ для определения растений. К.: Наук. думка, 1964. 36 с.
5. Азизов З.К., Пьянков С.А. Определитель минералов: Учебное пособие. Ульяновск: УлГТУ, 2006. 53 с.
6. Юбельт Р. Определитель горных пород. М.: Мир, 1977. 260 с.
7. Музафаров В.Г. Определитель минералов, горных пород и окаменелостей. М.:Недра, 1977. 327 с.
8. Сайт Бондарева Всеволода Петровича URL: <http://vsevolodbondarev.com/html/minerals.htm> (дата звернення 1.03.2019).
9. Сайт о минералах, свойствах камней, легенды о камнях, магия камней. URL: <http://world-of-stones.ru> (дата звернення 2.03.2019).
10. MongoDB 1.0 GA первый релиз новой СУБД. URL: <https://www.nixr.ru/news/9901.html> (дата звернення 10.03.2019).
11. Опис середовища розробки Visual Studio Code URL: <http://www.opennet.ru/opennews/art.shtml?num=42132> (дата звернення 01.04.2019).
12. JavaScript URL: <https://uk.wikipedia.org/wiki/JavaScript> (дата звернення 20.04.2019).
13. Документо-орієнтована система керування базами даних URL: https://uk.wikipedia.org/wiki/Document-oriented_database (дата звернення 18.04.2019).
14. Getting Started React Native. URL: <http://reactnatedocs.ru/docs/getting-started.html> (дата звернення 24.05.2019).