

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Құлманов Дәулетхан Полатұлы

Қайталанатын файлдарды табуға арналған мобильді қосымша жасау

ДИПЛОМДЫҚ ЖҰМЫС

5B070300 – «Ақпараттық жүйелер» мамандығы

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

КАӨЖС кафедрасы меңгерушісі

техн. ғыл. канд., ассистент-

профессор

 Н.А. Сейлова

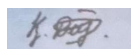
«27»мая 2021 ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: «Қайталанатын файлдарды табуға арналған мобильді қосымша жасау»

5B070300 – «Ақпараттық жүйелер» мамандығы

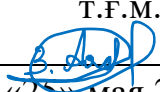
Орындаған



Құлманов Д.П

Ғылыми жетекші

т.ғ.м., лектор

 Зиро А.А.

«25»мая 2021 ж.

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

5B070300 – Ақпараттық жүйелер

БЕКІТЕМІН

КАӨЖС кафедрасы меңгерушісі
техн. ғыл. канд., ассистент-
профессор

 Н.А. Сейлова
« 27 » мая 2021 ж.

**Дипломдық жұмысты орындауға
ТАПСЫРМА**

Білім алушы: Құлманов Дәулетхан Полатұлы

Тақырыбы: «Қайталанатын файлдарды табуға арналған мобильді қосымша жасау»

Университет Ректорының 2020 жылғы «24» қараша №2131-б бұйрығымен бекітілген.

Орындалған жұмыстың өткізу мерзім «__» _____ 2021 ж.

Дипломдық жұмыстың бастапқы мәліметтері: Берілген тақырып бойынша әдебиеттерге шолу кезінде жиналған мәліметтер, теориялық материалдар жинау.

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

1. Тақырып аясын талдау және шолу;
2. Техникалық тапсырманы әзірлеу;
3. Бағдарламалық қамтаману құру;


Графикалық материалдардың тізімі (міндетті түрде қажет сызбалар көрсетілген): жұмыстың _____ слайдтан тұратын презентациясы көрсетіледі.

Ұсынылған негізгі әдебиет 10 кітаптан тұрады.

Дипломдық жұмысты даярлау
КЕСТЕСІ

Бөлім атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімі	Ескерту
Тақырып аясын талдау және шолу	1.01.2021 – 08.02.2021	
Техникалық тапсырманы әзірлеу	10.02.2021 – 3.03.2021	
Бағдарламалық қамтаманы құру	5.03.2021 – 28.04.2021	

Дипломдық жұмыс бөлімдерінің кеңесшілері мен норма бақылаушының аяқталған жұмысқа қойған
қолтаңбалары

Бөлімдердің атауы	Кеңесшілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қол қойылған мерзімі	Қолы
БҚ жасау			
Норма бақылаушы	М.А.Кабдуллин, тьютор		

Ғылыми жетекшісі  Зиро А.А.

Тапсырманы орындауға қабылдаған білім алушы  Құлманов Д.П.

Күні «25» мая 2021 ж.

АҢДАТПА

Менің жасаған дипломдық жұмысымды, күнделікті өмірде, барлық салалар аясында қолдануға болады. Себебі: бұл мобильдік қосымшаны кез келген адам, кез келген сәтте жүктеп алып, пайдалана алады. Мобилдік қосымша барлығы білетін Java бағдарламалау тілінде, Android Studio платформасында жасалды. Дипломдық жұмыстың мақсаты - қайталанатын документ, файлдарды, медиа- файлдарды табуға, оларды өшіруге мүмкіндік беретін қосымша құру. Дипломдық жұмыстың әр тарауы тақырыптың қазіргі жағдайын және оның өзектілігін көрсетеді. Жоба клиенттерді, оның ішінде медиа файлдармен, басқа да форматтағы файлдармен жиі іс жүргізетін жандарға арналған.

АННОТАЦИЯ

Моя диссертация может быть использована в повседневной жизни во всех сферах. Это мобильное приложение может быть загружено и использовано кем угодно в любое время. Мобильное приложение создано на всем известном языке программирования Java, платформе Android Studio. Целью дипломной работы является создать приложение, которое позволяет находить дубликаты документов, файлов, медиафайлов и удалять их. Каждая глава диссертации отражает текущее состояние темы и ее актуальность. Проект рассчитан на тех, кто часто имеет дело с клиентами, в том числе с медиафайлами и файлами других форматов.

ANNOTATION

My diploma can be used in everyday life in all spheres. This mobile application can be downloaded and used by anyone at any time. The mobile application is created in the well-known Java programming language, the Android Studio platform. The aim of the thesis is to create an application that allows you to find duplicate documents, files, media files and delete them. Each chapter of the thesis reflects the current state of the topic and its relevance. The project is designed for those who often deal with clients, including with media files and files of other formats.

МАЗМҰНЫ

КІРІСПЕ	6
1 Ақпараттық жүйе туралы жалпы түсінік	7
1.1 Ақпараттық жүйе құру өзектілігі	7
1.2 Ақпараттық жүйелердің қасиеттері	7
2 Пәндік аймақты талдау	9
2.1 Тақырыптың өзектілігі	9
2.2 Есептің қойылымы	9
3 Қосымшаның құрылымын жобалау	11
3.1 Қосымшаның концептуалды моделін құру	11
3.2 ER-диаграмма	12
3.3 Мәліметтер қорына жалпы шолу	13
3.4 Мәліметтер базасының кестелерін талдау	14
4 Қосымшаны жүзеге асыру	16
4.1 Пәндік аймақтың бағдарламалық жасақтамасын құру	16
4.2 Қосымша құрылымы	18
4.3 Жұмыстың сипаттамасы	20
ҚОРЫТЫНДЫ	24
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ	26
А ҚОСЫМШАСЫ	27
В ҚОСЫМШАСЫ	32

КІРІСПЕ

Қазіргі кезді электронды ақпарат заманы деп атасақта болады деп ойлаймын, өйткені қазіргі таңда технологиялар қоғам өмірінің барлық салаларына терең еніп, адамдардың уақытты өлшемді қолдана отырып, адам өміріне өте қарқында енуімен куәгерміз. Әсіресе 2020 жылдан басталған пандемия заманында, барлық жер карантинға жабылып, онлайн жұмыс атқаруы, ақпараттық заман болуына жаңа серпім берді.

Бүгінгі күнде еңбектеген баладан, еңкейген кәріге дейін, әр қайсысында гаджеттеріне тәуелді, деп айтуға да болады. Күнделікті өмірде ұялы телефонды, компьютерді немесе планшетті пайдаланбайтын адам жоқ. Тіпті қарапайым сағатымызда электронды болып, тек сағат қарауға емес, басқа да мүмкіншіліктерге толы заман болды. Бұл ақпараттың көбеюіне әкеліп соғып жатқан себепті, гаджеттегі жадыны үнемдеуге тура әкеліп соғып жатыр.

Бұл технологиялардың көпшілігі адам өміріндегі қызметтерді оңтайландыруға, білім алуға және уақытты үнемдеуге мүмкіндік береді. Қазіргі таңда барлық іс-шара, кездесулер онлайн түрде болып жатқандықтан, ұялы телефонымызда көп файлдар сақтау ыңғайлы. Алайда, ол файлдар жадыда көп орын алып, құрылғының жұмысына кері әсерін тигізуі мүмкін. Сол үшін ұялы телефондарды қайталанатын файлдардан тазалайтын қосымша бүгінгі күні сұранысқа ие.

Күнделікті өмірде смартфон кездейсоқ фотосуреттер мен бейнелерді түсірудің және құжаттар жасаудың танымал құралы болып табылады. Бірақ көп ұзамай біздің құрылғылар қажет емес селфиермен, құжаттармен, видеолармен толып, жұмыс істеуі баяулайды. Телефондағы қайталанатын файлдар көп орын алады және оларды жою керек. Бұл «Duplicate master» бағдарламасы сізге қайталанатын файлдарды анықтау үшін телефонды жедел қарап шығуға көмектеседі және телефонның жадын, жылдамдығы мен өнімділігін арттыру үшін оларды тексеруге және жоюға мүмкіндік береді.

Бұл дипломдық жұмыстың тақырыбын таңдауға, осы қосымшаны жасауға итермелеген, өзімнің медиа-файлдармен, фотосуреттермен жиі жұмыс жасауым және құрылғының жадысын жиі босатып тұру қажеттілігімен көп соқтығысуым себеп болды.

Жұмыстың әр бөлімінде тақырыптың өзектілігі және қазіргі уақыттағы қолдану барысы көрсетілген. Нәтижесінде мен өзіме және кез келген саладағы жандарға көмектесетін Мобилдік-қосымшанысын жазып шықтым.

1. Ақпараттық жүйе туралы жалпы түсінік

1.1 Ақпараттық жүйе құру өзектілігі

Ақпараттық жүйе- негізгі ұйымның белгілі бір тұлғаның , компанияның,қысқаша айтқанда қандай да бір ұжымның өзіне керекті мәліметті алуға,сақтауға,жіберуге,өңдеуге тағы да басқа іс-әрекеттер жасауға мүмкіндік беретін жүйе. Бұл процестерден басқа, ақпараттық жүйелер ақпаратты тазарту, нақтылау және қорғау сияқты процестерді жүзеге асыра алады.

Қазіргі таңда ақпараттық жүйе түсінігін ноутбукта немесе компьютерде ақпаратты пайдаланудың негізгі техникалық құралдары ретінде көреді. Үлкен компаниялар бұл қызметтерге мэйнфрейм,супер электрондық есептеуіш машинасын да қолданысқа енгізген.Дегенмен адами фактор болмаса, ақпараттық жүйелерді техникалық түрде іске асырудың қажеттілігі бар дегенге келіспеймін.

Ақпараттық жүйелердің даму кезеңі. 1960 жылдары ақпараттық жүйеге елдің көзқарасы өзгере бастады. Бірнеше параметрлері бар периодты есептер одан алынған ақпараттар қолданысқа ене бастады. Ол үшін халыққа көп функцияларды қамсыздандыра алатын,мүмкіндігі мықты компьютерлік құралдар қажет болды.

1970 жылдары шешімді тез шешу мақсатында жағдайды жылдамдату үшін ақпараттық жүйелерді басқаратын машиналарды көптеп халық қолданысқа енгізді.

1980 жылдарды ақпараттық жүйелер концепциялық қолданыста әрімен өзгеруіне әкеліп соқты. Олар ақпараттың стратегиялық қайнар көзі және барлық саладағы,бағыттағы адамдар пайдалана алатын мүмкіндікке ие болды. Осы жылдардағы ақпараттар жүйесі ақпаратты уақытылы жеткізіп, компаниялардың өз жұмысы бойынша дамуына, алға жылжуына үлкен әсерін тигізді.

1.2 Ақпараттық жүйелердің қасиеттері

Қандайда бір ақпараттық жүйе өзінің формасын, келесі жүйенің барлық формасы негізі бойынша анализ өтуіне сотығысу болжамы бар. Ақпараттық жүйелер динамикалық және дамушы бола алады.

- Ақпараттық жүйелерді жасау кезеңдерінде жүйелік көмек жасау қажет
- Ақпараттың жүйенің пайда болу өнімдері шешім бойынша ақпараттар бола алады.

Қазіргі кезде ақпараттық жүйелерді дербес компьютердің арқасында таралған жүйелер десті. Дегенмен ақпараттық жүйені дербес компьютерсіз вариантта алып көруге болады. Көптеген компаниялар тауарларды сату,сатып алу процесінде ақпараттық жүйе тұтынушыны тіркеуге алады. Ол:

- тұтынушылар тобын анықтап, олардың сұранысын және құрамы, одан соң топтарға қарай.
- жиі сатып алатын клиенттерге әр-түрлі тиімді ұсыныстар тастауға.
- тұрақты тұтынушыларға тауарларға рұқсат береді.

Ақпараттық жүйедегі процесстер. Ақпараттық жүйе жұмысын жүзеге асыру процессін блокта істелген схема түрінде қарауға мүмкіндік бар болады.

- Жүйенің ішкі және сыртқы көздерден жазу;
- Жазылған ақпаратты ыңғайлы түрмен көрсетіп және өңдету;
- Клиентке қажетті ақпаратты шығару.
- Кері байланыс бұл кіру ақпаратының фильтр жасау болғандықтан сол компанияның жұмысшыларына өңделген ақпарат.

2. Пәндік аймақты талдау

2.1 Тақырыптың өзектілігі

Қазіргі таңда смартфондар кездейсоқ фотолар мен бейнелерді түсірудің және құжаттар жасаудың танымал құралдарының бірі болып табылады. Бірақ көп ұзамай біздің құрылғылар қажет емес суреттермен, видеолармен, құжаттармен толып, ұялы телефонның жұмыс істеуіне кері әсерін тигізеді. Телефондағы қайталанатын файлдар көп орын алады және оларды жою керек. Сол себепті, пәндік аймақ ретінде- қайталанатын файлдарды тауып, өшіруге арналған қосымшаны алдым.

Қазіргі пандемия заманында барлық іс-шара, кездесулер онлайн түрде болып жатқандықтан, ұялы телефонымызда көп файлдар сақтау ыңғайлы. Бірақ та, ол файлдар жадыда көп орын алып, құрылғының жұмысына кері әсерін тигізуде. Сол үшін ұялы телефондарды қайталанатын файлдардан тазалайтын қосымша бүгінгі күні өте қажет деуге келеді. Қосымша Android Studio платформасына жасалынды.

2.2 Есептің қойылымы

Дипломдық жұмыстың мақсаты- қайталанатын документ, файлдарды, медиа- файлдарды табуға, оларды өшіруге мүмкіндік беретін қосымша құру.

- Жүйеде келесідей функциялар орындалуы керек:
- Барлық қолданушыға бірдей қолжетімділікті ұйымдастыру
- Табылған мәліметтерді құпия(конфиденциальность) сақтау;
- Құрылғыға зиян келтірмеу;
- Қажет емес мәліметтерді азайтып, жадыда бос орын көбейту;
- Бірнеше рет қайталанатын кез- келген форматтағы файлды таба алады;
- Табылған файлды өшіру мүмкіндігі бар;
- Қолданысқа ыңғайлы интерфейс;
- Жадыны үнемдейді;
- Пайдалану оңай болуы қажет.
- Қосымшаны қамтамасыз ету,мәзір қаншалықты әдемі болса да адамдар оны пайдаланатынын ұмытпағанымыз жөн.
- техникалық қолдаудың болуы.
- Барлық мобильдік жүйе бірқалыпты толықтыруды мен жаңарту қажет етеді.

Жобаның құнына келетін болсақ. Қосымша барлығына қолжетімді,себебі:мобильдік қосымша тегін,тек қазіргі уақытта андройд уялы телефоны болса болды.

Бұл жұмысым кез кезген,тіпті жұмыссыз жанды алсақта,ол адам уялы телефоны арқылы суретке түседі,өлең тыңдайды және тағы да басқа әректтер жасайды. Бұның барлығы құрылғының жадысының толып,жұмысына кері әсерін тигізуіне әкеліп соқтырады.Осы кезде қолданушыға жадыны тез әрі қауіпсіз босатуға мүмкіндік бететін мобильдік қосымша.

3. Қосымшаның құрылымын жобалау

3.1 Қосымшаның концептуалды моделін құру

Ақпараттық жүйені құру және жобалау оны қолданудың концептуалды моделін құрудан басталады. Ақпараттық жүйені пайдаланудың концептуалды моделі ең алдымен ақпараттық жүйе құру үшін нақты міндеттер мен функциялар ауқымын анықтайды. Мен дайындайтын қосымшаның концептуалды моделіне тоқталайық.

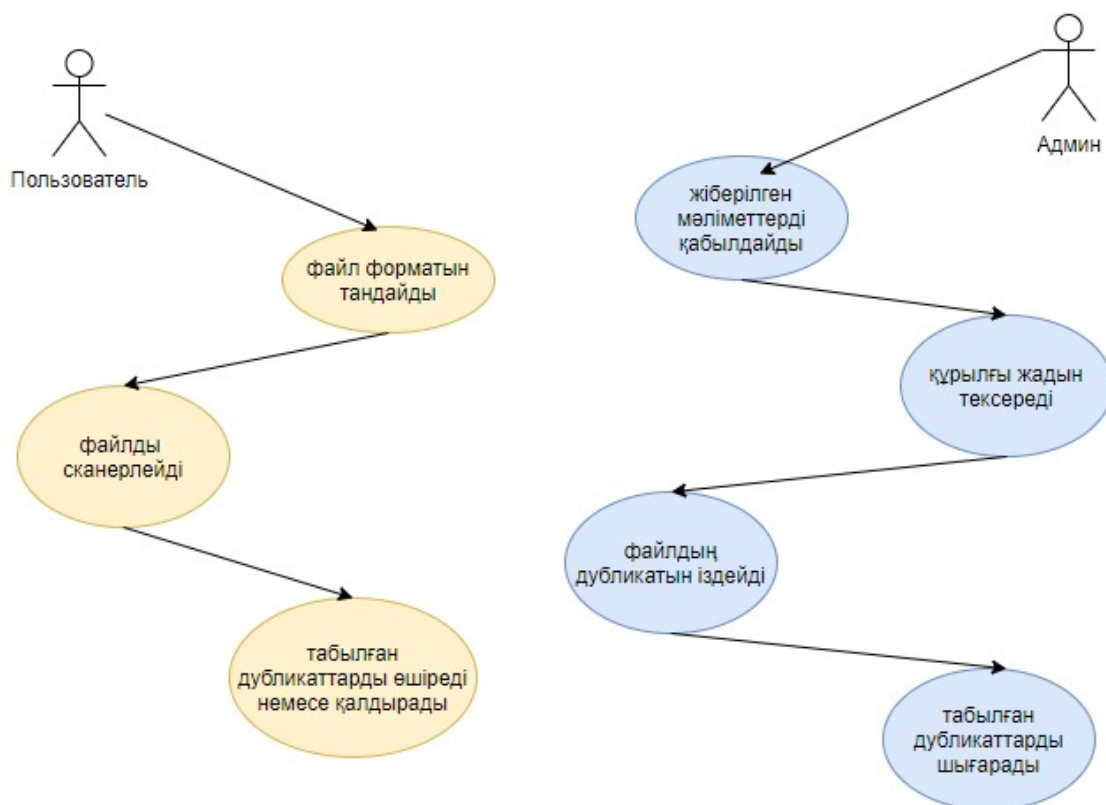
Бұл жерде прецеденттер диаграммасы көрсетілген. Қолданушы да, әкімші де өз кезегінде әртүрлі іс-әрекеттер орындайды.

- Қолданушыға тиесілі іс-әрекеттер:
- Файл форматын таңдау;
- Файлды сканерлеу;
- Табылған дубликаттарды өшіру немесе қажетіне жарату.

Әкімші өз кезегінде:

- Жіберілген мәліметтерді қабылдайды, өтейді;
- Құрылғы жадын тексереді;
- Қолданушы таңдаған формат бойынша құрылғыдан дубликатты іздейді;
- Табылған дубликаттарды шығарады.

Прецеденттер диаграммасының негізгі мақсаты- заказ берушіге, пайдаланушыға және жобаны әзірлеушіге бірлесіп жүйені талқылауға мүмкіндік беру және жүйенің функционалдылығын талдау, сипаттау.



Сурет 3.1-Прецеденттер диаграммасы

3.2 ER-диаграмма

«Субъект-қатынас» диаграммасы(ER диаграмма)- бұл жүйе ішіндегі әр түрлі «объектілердің»(адамдар,объектілер, ұғымдар және т.б.) өзара байланысты екендігін көрсететін блок-схеманың түрі. ER диаграммалары әр түрлі саладағы реляциялық мәліметтер базасын жобалау және түзету, бизнеске арналған бағдарламалық және ақпараттық жүйелерді зерттеу және әзірлеу үшін қолданылады. ER диаграммалары нысандарды, олардың атрибуттары мен байланыстарын бейнелеу үшін стандартты белгілер жиынтығына сүйенеді.

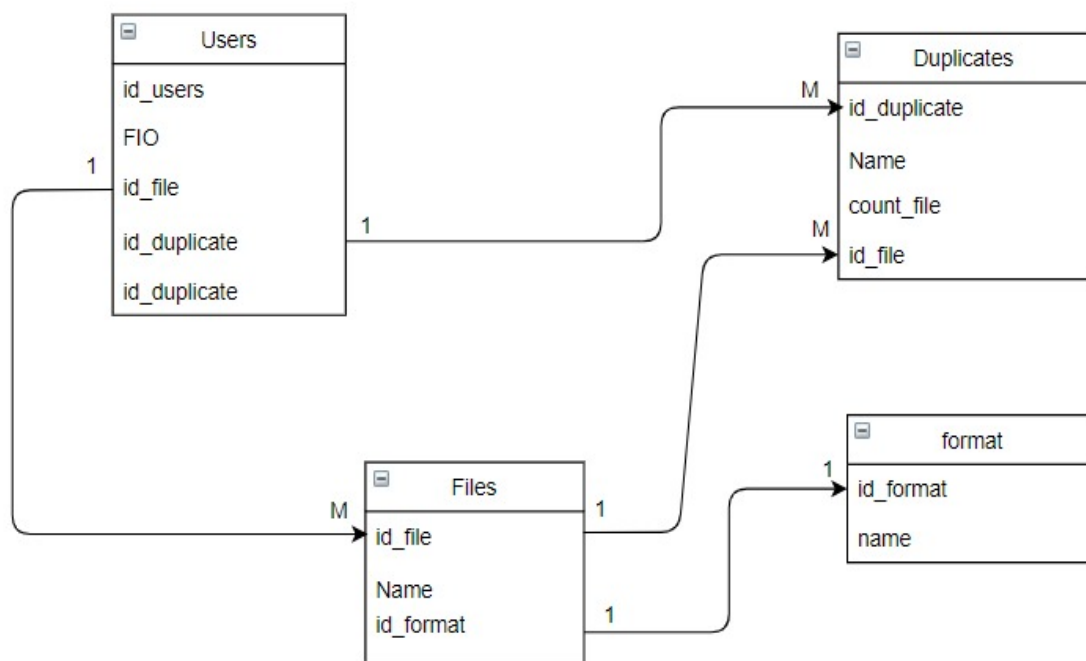
Негізгі кілт- бұл әрбір жазба үшін ерекше идентификатор, оның мәні тек сол жазбаны бірегей анықтайды.

Негізгі кілт екі қасиетке ие болуы керек:

- Жазбаны бірімәнді сәйкестендіру;
- Артық(избыточность) болмауы-бұл әрбір жазба үшін ерекше идентификатор, оның мәні тек сол жазбаны бірегей анықтайды.

Жобаланған мәліметтер базасында барлық кестелер үшін басты кілт арнайы енгізілген идентификатор(id) болып табылады, яғни,біздің жүйеде:

- Users кестесі үшін users_id;
- files кестесі үшін files_id;
- duplicates кестесі үшін duplicates_id;
- format кестесі үшін format_id.



Сурет 3.2 -ER- диаграмма

3.3 Мәліметтер қорына жалпы шолу

Қандайда қосымша құру кезінде таңдалған пәндік аймақ бойынша мәліметтер базасы құру қажет. Мәліметтер базасы- қандай да бір структураға қатысты мәліметтерді сақтау баяндау үшін қарастыратын белгілі бір ережелерге сәйкес біріктірілген мәліметтер жиынтығы. Басқаша айтқанда, мәліметтер қоры- ақпаратты сақтауға арналған құрылым.

Қарапайым мысалдар: банктен кредит алушылар кестесі, ұшақтар қозғалысы кестесін, кәсіпорын туралы мәліметтерді, студенттердің оқуға түсу туралы қорытындылары және тағы басқаны атауға болады. Әдетте, мәліметтер базасы қандай да бір мәселенің шешімін табуға арналған файлдар кестесі немесе кесте түрінде ұсынылады.

Мәліметтер қорының концепциясы деректер қорын басқару жүйесінің концепциясымен тығыз байланысты. Деректер қорын басқару жүйесі(ДҚБЖ)- дегеніміз жаңа мәліметтер базасының құрылымын құруға, оны толтыруға, бар ақпараттарды визуалдауға арналған бағдарламалық құралдар жиынтығы.

Мәліметтер қорының негізгі элементі- кесте. Мәліметтер қорының кестелерінің бағандары өрістер, ал жолдары жазбалар деп аталады. Деректер қорының кестесін құру барысында ең алдымен оның құрылымын анықтау қажет. Келесі кезекте- кестеге жазба енгізу,оны реттеп өзгерту. Бірақ кесте жолдары бос болса да құрылған болып саналады.

Мәліметтер базасында көрсетілген барлық мәліметтердің құрылымы бұл маңызды элемент болуы керек, өйткені ол пәндік аймақты көрсетеді. Барлық мәліметтер қасиеттерінің құрылымы арасында қисынды байланыс болады.

Мәліметтерді тиімді өңдеу үшін бастапқы бірнеше схема аз бөліктерге бөлінген. Әрбір бөлік түпнұсқа схеманың белгілі бір мағынасы бар қамтиды. Осы бөліктердегі жекелеген ішкі схемалар арасындағы байланыс үшін сол өрістерді пайдалану қажет.

Кестелердегі өрістер оның құрылымын және ұяшықтарға жазылған деректердің топтық қасиеттерін анықтауға мүмкіндік береді. Мәліметтер қорының негізгі қасиеттері:

- Өріс атауы – таңдалған өріске сілтеме жасау үшін қолданылады.
 - Өріс типі - сол өрісте болуы мүмкін мәліметтер типін анықтайды (мәтін, сандық, даталар және т.б.).
 - Өріс өлшемі - өріске сәйкес келетін деректердің максималды ұзындығын анықтайды.
 - Өріс форматы - бұл ұяшықтардағы деректерді форматтау әдісі.
- Кестелер мен сұраныстар бір-бірімен схема арқылы байланысады.

ДҚБЖ әзірлеу технологиясы бірнеше кезендерді қамтиды, оның ішінде негізгілері:

- Мәліметтер қорының құрылымын жобалау;
- Кестелер арасындағы байланысты жобалау;
- Әр кестенің жеке құрылымын жасау;
- Кестелерге мәліметтерді енгізу;
- Сұранысты өңдеу;
- Кестелер мен сұраныстар арасындағы қатынастарды жүзеге асыратын мәліметтер схемаларын әзірлеу;
- Мәліметтер базасын басқаруға арналған программалық модульдерді дайындау;
- Құжаттарды шығаруға арналған есептерді дайындау.

Бұрынғы кезендердің дәлділігі қатал емес. Негізінде ДҚБЖ әзірлеушісі жобаны нақтылай отырып, сол кезендерге орала алады.

3.4 Мәліметтер базасының кестелерін талдау

Құрылатын бағдарламаның мәліметтер базасын құру қажет. Ең бірінші кезекте жобалау барысында келесі кестелерді құрамыз:

USERS(Қолданушылар)(Id_Users, FIO, Id_file, Id_duplicate)

FILES(Id_file, Name, Id_format)

DUPLICATES(Id_duplicate, name, count, id_file)

FORMAT(Id_format, name)

Негізгі кілттік өрістер: Id_Users, Id_file, Id_duplicate, Id_format.

Кесте атауы	Өріс атауы	Мәліметтер типі
Users	<u>Id_Userrs</u>	<u>int</u>
	FIO	<u>Nvarchar(100)</u>
	<u>Id_file</u>	<u>int</u>
	<u>Id_duplicate</u>	<u>int</u>

Кесте 3.1- Users

Кесте атауы	Өріс атауы	Мәліметтер типі
Files	<u>Id_file</u>	<u>int</u>
	name	<u>Nvarchar(50)</u>
	<u>Id_format</u>	<u>int</u>

Кесте 3.2- Files

Кесте атауы	Өріс атауы	Мәліметтер типі
Duplicates	<u>Id_duplicate</u>	<u>int</u>
	name	<u>Nvarchar(50)</u>
	count	<u>int</u>
	<u>Id_file</u>	<u>int</u>

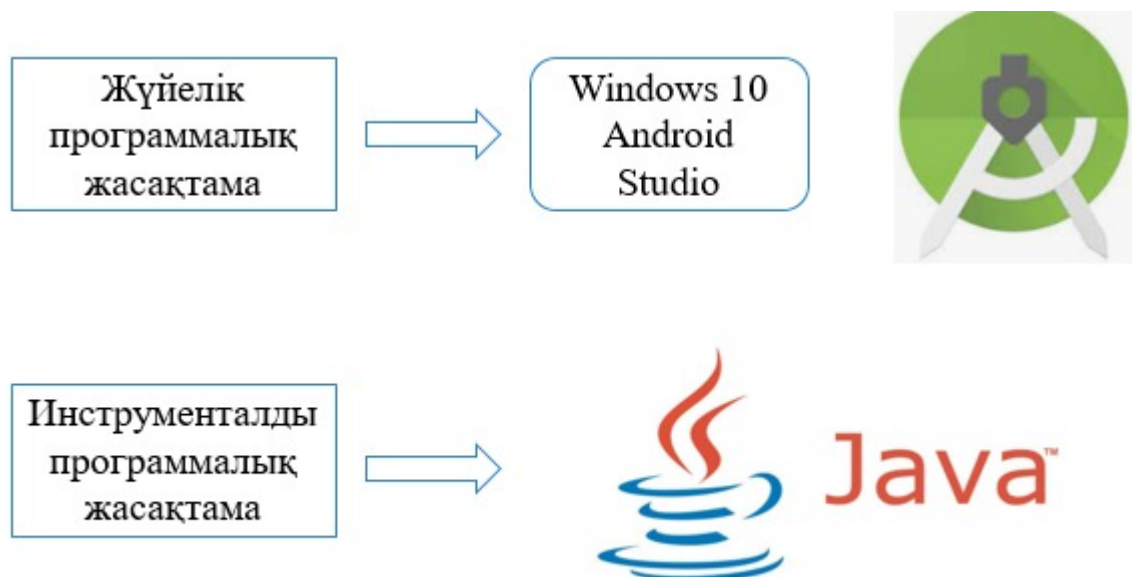
Кесте 3.3 - Duplicates

Кесте атауы	Өріс атауы	Мәліметтер типі
format	<u>Id_format</u>	<u>int</u>
	name	<u>Nvarchar(50)</u>

Кесте 3.4 –Format

4. Қосымшаны жүзеге асыру

4.1 Пәндік аймақтың бағдарламалық жасақтамасын құру



Сурет 4.1-Қосымшаның бағдарламалық жасақтамасын құру үшін қолданылатын құралдар

Android дегенді біз ұялы телефонға байланысты операциялық жүйелердің бірі екенін білеміз. Android жүйесі өзінің қосымшалар жазуға, даярлауға ыңғайлы жабдық жиынтығын ұсынады. Бұндай жиынтық немесе жабдықтар белгілі қосымшаны құрайды Android Studio, ол қосымшаны әзірлеуге арналған ресми Android IDE. Android Studio әр түрлі тілдерді қабылдай береді. Олар: Котин, Скала, Гроови, Java және Ключ. Менің жасаған жұмысым дәл осы тілдердің арасындағы Java тілінде жазылған.

Android студияда қосымша құруға мүмкіндік беретін әртүрлі компоненттер бар. Gradle негізінде құру жүйесі, нұсқа құру және бірнеше APK файлдар, сондай-ақ қосымшаны құруға көмек беретін код шаблондарда бар. Элементтердің тақырыптарын сүйреп апару, өңдеуді қолдайтын отырғызу редакторы. Пайдаланушыға қарапайымдылығы, код ProGuard қысқарады және Gradle аз жұмсайды.

Android студияда бағдарламаларды жүзеге асыруды қолға алатын даму кезеңдерінің ішінде біз 4 кезенді табамыз.

– Біріншісі қоршаған орта параметрлері. Бұл кезеңде даму ортасы конфигурацияланады және орнатылады. Сонымен қатар, бағдарламаны орнатуды жүзеге асыруға болатын элементтерге қосылады және Android виртуалды жабдықты (AVDS) жасалады.

– Екінші кезең Жобаның конфигурациясы және дамуы. Бұл уақытта жобаның оны әзірлеу және конфигурациясы жүзеге асырылады. Біз программаның және бастапқы код файлдарының ресурстарын қамтитын модульдерді жасау туралы айтып отырмыз.

– Үшінші кезең мыналарды қамтиды қолданбаны тестілеу, күйін құру және келтіру. Осы кезде жұмыс эмуляторда немесе Android құрылғысында орнатылатын және іске қосылуы мүмкін .apk бумасына түзеледі. Gradle негізделген құрастыру жүйесін қолданамыз. Осы икемділікті, құрастырудың өзгертілген тәуелділікті және нұсқаларын жасауды қамтамасыз етеді. Өзге IDE қолданған жағдайда, жобаны Gradle көмегімен жасауға болады және өз кезегінде АДБ қолданатын уялы телефонға қондырылады. Кейіннен бағдарламаны бақылау құрылғының бақылау хабарламалары және IntelliJ идеясымен бірге Android тіркеу құрылғысы бойынша түзетіледі. Сонымен қатар, Android SDK жүйесімен жасалған тіркеу және түзету құралдарын қосып, үйлесімді JDWP түзеткішін қолдануға болады. Соңында қосымшаны тексеру үшін Android SDK тест құралдары қолданылады.

Біз қазір Android бағдаоламасын құру дамуы мен кезеңдерін білеміз. Модульдік базаға бағыт жасай тұрып, әрбір жобаның бағдаоламасында басындағы код файлдары мен ресурстық файлдары бар бір немесе бірнеше модульдер бар. Android модульдері, тестілеу модульдері кітапхана модульдері және App Engine модульдері. Бастапқы бойынша, Android студия жоба файлдарын Android жоба көрінісінде көрсетеді. Осы жолы модульдер негізгі кодтық файлдарға тез қол жеткізуді қамтамасыз ететін ұйымдасқан түрінде көрсетіледі. Файлдар жасалған жағдайда, олар Scripts Gradle астында жоғарғы деңгейде көрінеді. Android Studio біз Gradle бағдарламалық қосымша жүйесінің негізі ретінде қолданылатынын білеміз. Осы құру жүйесі Android студия мәзірінде қосарланған құрал ретінде жұмыс жасайды және өз жолында пәрмен жолына тәуелсіз.

Java расында да желілік ортада мықты құрал болып саналады. Дегенмен ол тек онымен шектеліп қалмай, Java тілі көптеген бағдарлама жасау үшін пайдаланылатын универсалды тіл болып табылады. Бұған қоса Java платформаға тәуелсіз проекттер жасауға да қолданылады. Программалаудың қауіпсіздігі және қарапайымдылығын амалдары жылдам және түсінікті код жазуға мүмкіншілік береді. Өзге бағдарлау тілдерінде кеңінен тараған қателер Java қосымшаларында болмайды. Дегенмен, барлық ішкі әдіс-амалдарына қарамай Java бағдарлау тілі қарапайым, үйренуге оңай тіл.

Артықшылықтары:

- Архитектураға тәуелсіз Интернет программа үшін мінсіз.
- Басында жазылған код басқа жерде орындалады.
- Бірнеше ағымды қосымшалар желі арқылы тапсырманы орындайды.
- Динамикалы код қажет болған кезде ғана қолданылады.
- ООП Java программалау тілі қазіргі ООП негізіне сәйкес қосымшалар жазу үшін қолданылады.
- Жад басқарылады. Жад автоматты түрде бөлінеді.
- Garbage collector қолданылмайтын жадты өшіріп отырады.

– Бір-бірімен жақсы қиылысқан тілдің элементтері оның қолданылуын жеңілдетеді.

Java программалау тілі, ол JDK (Java Development Kit). Демек, программалау тілі бір-бірімен топтасқан бірнеше майда класстардан, бөлшектерден, тұрады. Яғни, Javaда жазылған қосымшалар, негізінде, осы класстардың жиынтығын, пакеттерді, қолданады. Жалғастырсақ, Javaда программалау кезінде қолданылатын процесстер:

- Жаваның стандартты пакетіндегі класстар
- Басқалардың халық үшін жазған класстары
- Өзіміз жазған класстар
- Javaда қосымшаның жазу процессі:

Жазу. .java файлын жазу арқылы өзіміздің классты анықтауға болады. Компиляция: жасалған .java файлды Java компиляторына жіберіп .class файлды алу арқылы. Орындау. пайда болған .class файлды Java интерпретаторына кодты жасау үшін жібереміз.

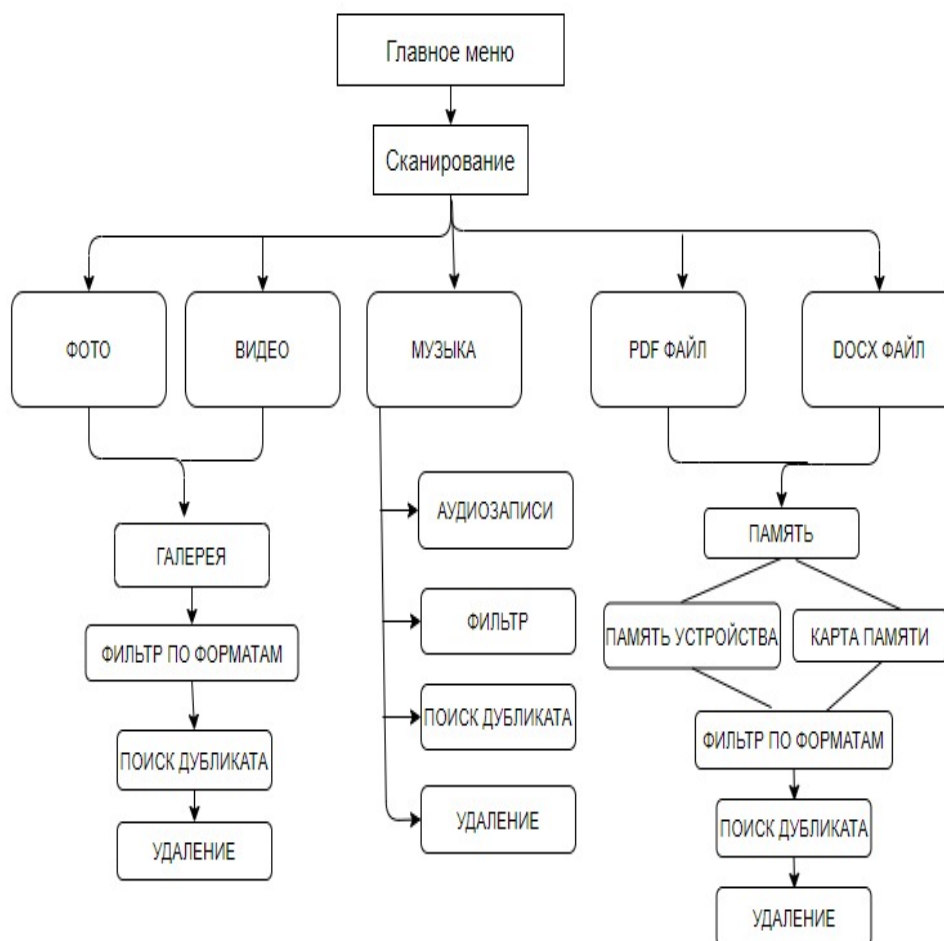
Java компиляторы – жасалған кодты жүзеге асыру үшін дайындайды және байт-коды бар .class файлдарын көрсетеді. Компиляция кезінде ақаулар болса, онда кодтағы ақауларды түзетіп қайтадан компиляциядан өткізу қажет. Байт-код – Java виртуалды машинасында енгізілетін оңтайланған нұсқаулардың тобы. Демек, JVM байт кодтың интерпретаторы. Осы әдіс басында күнделіктегіден тыс болып көрінуі мүмкін, себебі қазіргі күндегі программалау тілдерінің компиляторлары байт кодты шығармай, бірден орындалатын кодты шығарады. Бұндай әдістің өзіндік артықшылықтары бар және жылдамдық ерекшеліктері жақсы деуге келмейді. Java интерпретаторы .class файлды өткізіп оны дербес компьютерге түсінікті тілге өзгертіп, яғни қосымшаны орындайды.

4.2 Қосымша құрылымы

Қайталанатын және ұқсас файлдарды өшіретін қосымшаға кірген кезде басты бет пайда болады, басты бетте файл типтері бойынша қосымша жұмысын оңтайландыратын бірнеше бөлімшеге бөлінген:

- Аудиоларды сканерлеу;
- Видеоларды сканерлеу;
- Суреттерді сканерлеу;
- Документтерді сканерлеу;
- Толық сканерлеу;

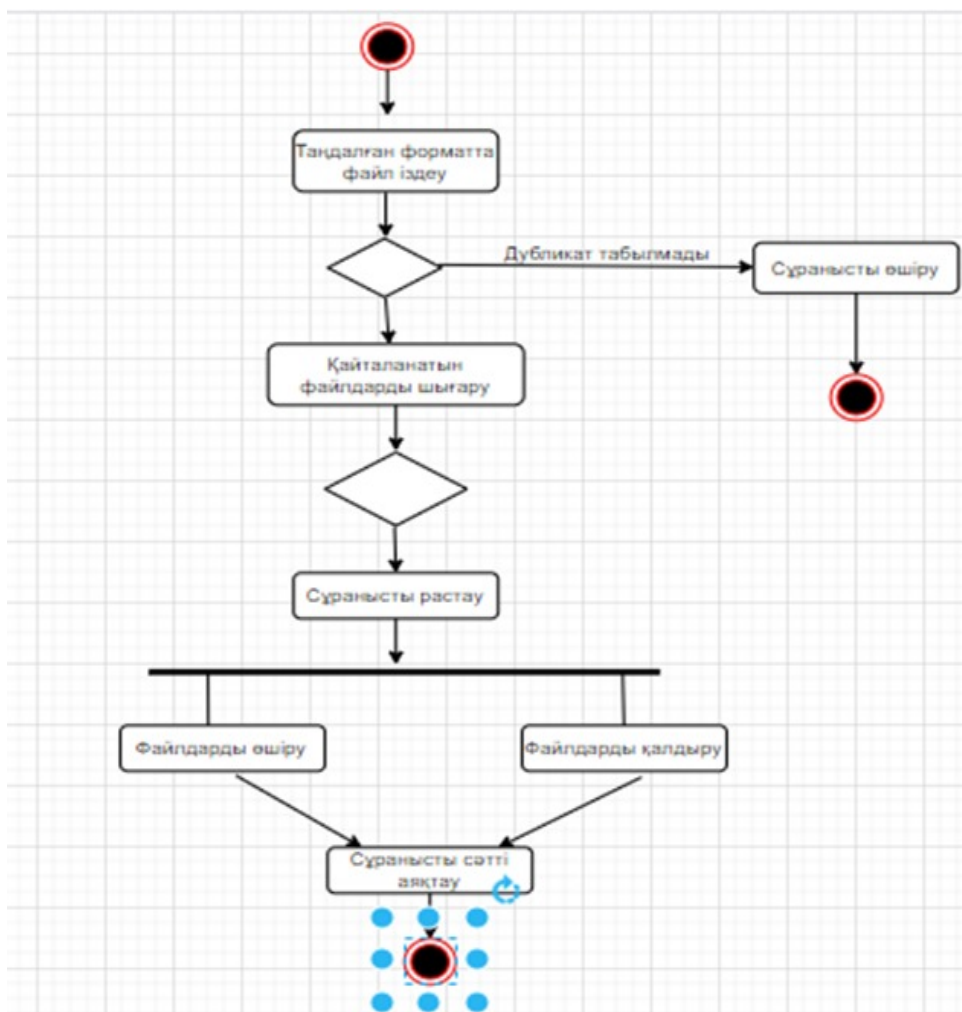
Таңдалған формат бойынша сканерлеу жүреді, яғни құрылғының жадындағы барлық файлдарды тексеріп, ұқсас немесе қайталанатын файлдар тізімін шығарады. Табылған дубликаттарды қолданушы таңдай алады: өшіре алады немесе керегіне жаратады.



Сурет 4.2- Қосымша құрылымы

Іс-әрекеттер диаграммасы қосымшада орындалатын процесстерді көрсетуге, жүйелеуге арналған. Диаграмманы талқылайтын болсақ:

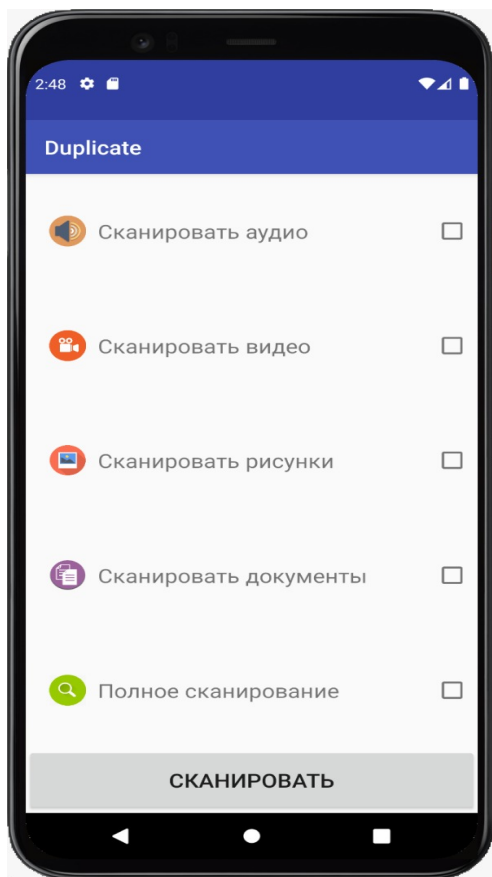
- Процесттің басталуы;
- Таңдалған форматта файл іздеу;
- Құрылғы жадын сканерлеу;
- Егер дубликат табылмаса, шешәмә жоқ, яғни процесс аяқталады.
- Егер табылса, ұқсас немесе қайталанатын файлдарды шығарады;
- Сұранысты растайды: Файлды өшіреді немесе жадыда қалдырады.



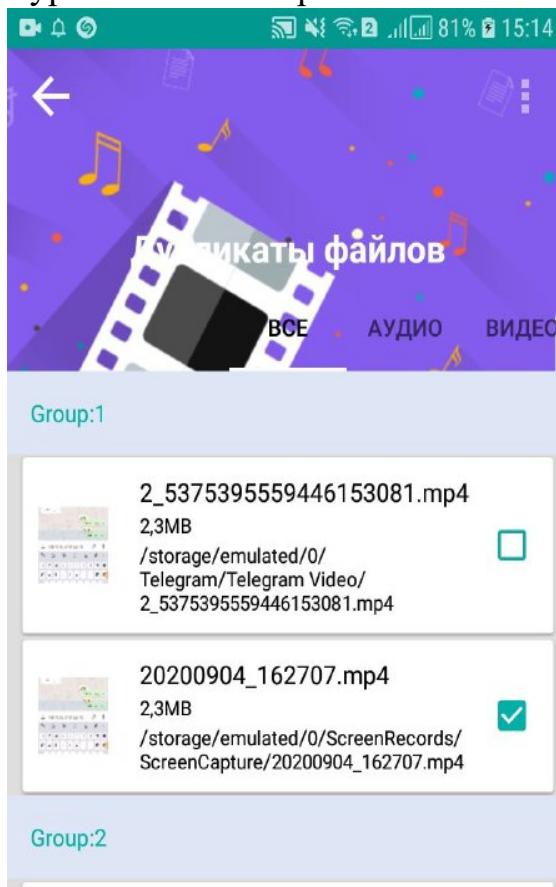
Сурей 4.3-Іс-әрекеттер диаграммасы

4.3 Жұмыстың сипаттамасы

Қосымшаны іске қосқанда бас мәзір ашылады. Бас мәзірде файл форматын таңдау мүмкіндігі бар. Таңдалған формат бойынша құрылғы жадын сканерлейді. Санерлеу барысында барлық ұқсас және қайталанатын файлдар тізімі шығады.

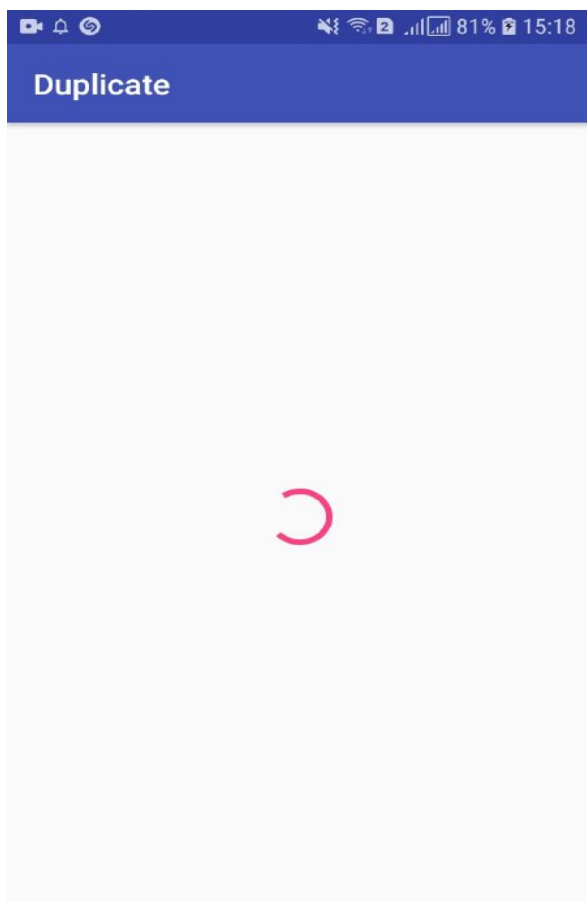


Сурет 4.4-Бас мәзір

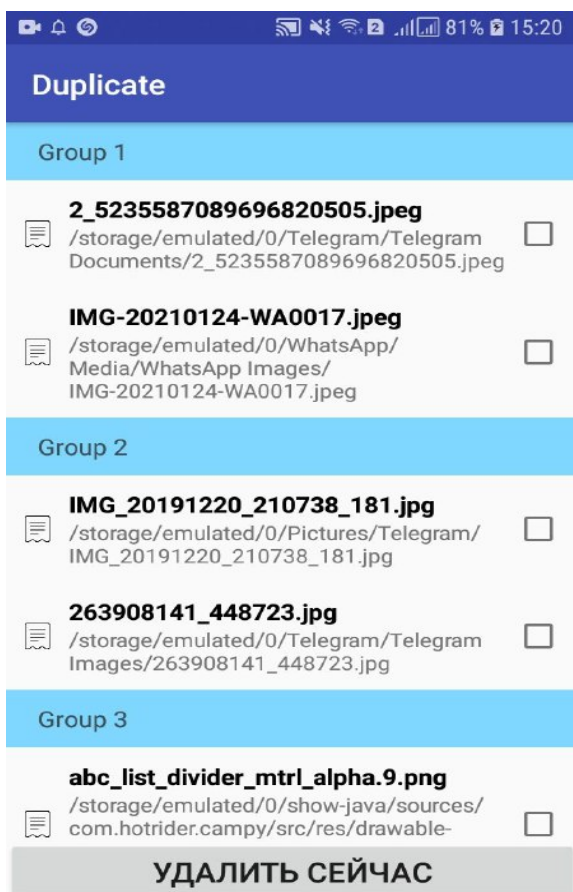


Сурет 4.5-Файл дубликаттары

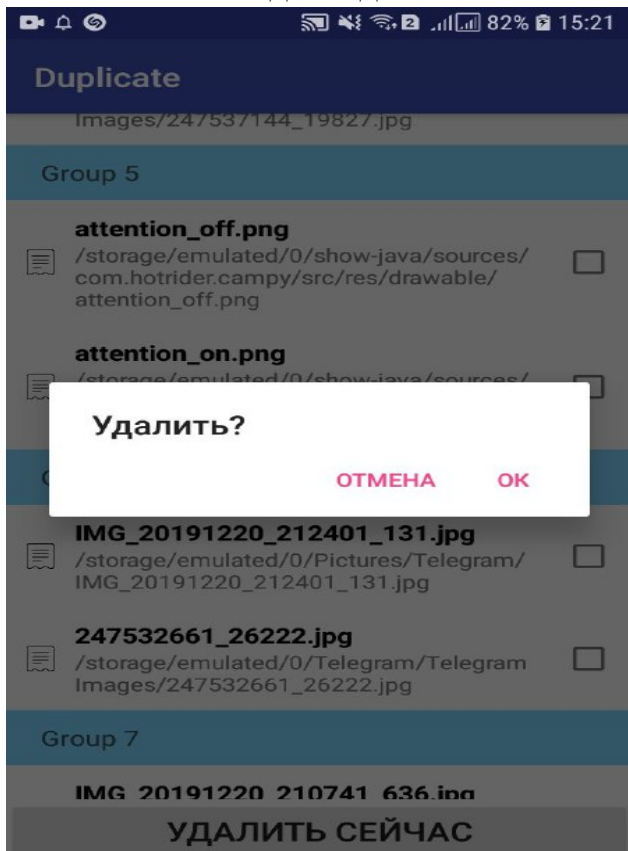
Жоғарыда 4.5-суретте көрсетілгендей құрылғыдан фотосуреттер сканерленген және қай суреттің дубликатын іздеу керекін таңдау жүзеге асуда.



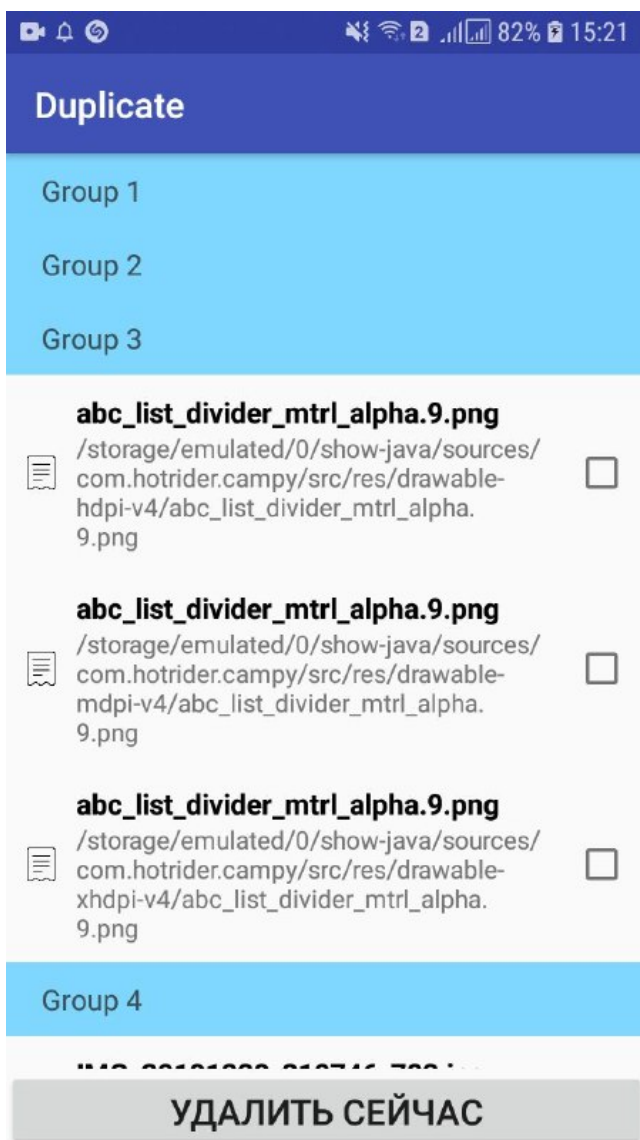
Сурет 4.6-Дубликатты құрылғы жадынан іздеу барысы.



Сурет 4.7-Таңдалған файл бойынша ұқсас және қайталанатын файлдар тізімі. Және осы бөлімде «Удалить сейчас» батырмасы да бар.



Сурет 4.8-Табылған дубликаттар арасынан қажет емес файлдарды таңдап сол жерде бірден өшіру мүмкіндігі бар.



Сурет 4.9-Өшірілген нәтиже бойынша жадыда 1,2,3 группаларда ұқсас немесе қайталанатын файлдар қалмады.

ҚОРЫТЫНДЫ

«Duplicate Master» барлық ұялы телефон қолданушыларға, әсіресе фотограф, видеограф, мобилограф, немесе құжаттармен көп жұмыс істейтін адамдарға арналған.

Бағдарламаның ең басты қызметі- бірдей немесе ұқсас атаулары бар және мазмұны бірдей файлдарды іздейді. Файлдың түрлі форматымен жақсы жұмыс істейді. Мысалы, музыка файлдарының әр түрлі тегтері болса да, бірдей музыкалық файлдарды таба алады. Сонымен қатар, фотосуреттерді салыстыра отырып, тек бірдей емес, жай ғана ұқсас фотосуреттерді таба алады.

Келесідей мүмкіндіктер береді:

- Барлық медиа файлдарды сканерлейді;
- Барлық қайталанатын деректерді, соның ішінде құжаттарды да, медиа файлдарды да өшіреді;
- Қауіпсіз іздеу үшін барлық анықталған қайталанатын файлдарды алдын ала қарап ала аласыз.
- Толық сканерлеу режимі және басқа кеңейтілген іздеу әдістері бар;
- Мультимедиа, құжаттар және басқаларға арналған мәзірді көрсетеді;
- Сізге қайталанатын жаңа файлдар туралы хабарландыру жіберіледі.

Қосымшаны жүзеге асыру үшін Android Studio ортасы және Java программалау тілі қолданылды.

Бастапқы дипломдық жұмыстың мақсаты- қайталанатын документ, файлдарды, медиа- файлдарды табуға, оларды өшіруге мүмкіндік беретін қосымша құру. Қойылған мақсатқа және міндеттерге қол жеткізілді:

- Барлық қолданушыға бірдей қолжетімділік бар;
- Табылған мәліметтерді құпия(конфиденциальность) сақталады;
- Құрылғыға зиян келтірмейді;
- Қажет емес мәліметтерді азайтып, жадыда бос орын көбейтеді.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Аллен М. E-learning: как сделать электронное обучение понятным [Текст]. / М. Аллен – Альпина Паблшер, 2016.
2. Вильямс. Разработка приложений клиент/сервер
3. Затонский А. В. Программирование и основы алгоритмизации / А.В. Затонский, Н.В. Бильфельд. – М.: Дрофа, 2014. – 176 с.
4. [https://kk.vvikipedla.com/wiki/Java_\(programming_language\)](https://kk.vvikipedla.com/wiki/Java_(programming_language))
5. https://kk.vvikipedla.com/wiki/Android_Studio
6. Варфоломеева А.О. Информационные системы предприятия: Учебное пособие / А.О. Варфоломеева, А.В. Коряковский, В.П. Романов. - М.: НИЦ ИНФРА-М, 2013. - 283
7. <https://malimetter.kz/intranet-tin-programmalau-kuraldary-zhane-tilderi/>
8. Байжұманов М.Қ., Жапсарбаева Л.Қ. “Информатика”
9. Семакин И.Г., Шестаков А.П. Основы программировани.-Москва, 2011г.
10. Г.С.Мухашева, И.И.Томилова, Б.О.Мухамеджанова, А.Н.Томилов «UML практикумы»

А ҚОСЫМШАСЫ

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical"
tools:context=".MainActivity">
<LinearLayout
android:id="@+id/layout_audio"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:gravity="center"
android:orientation="horizontal"
android:padding="10dip">
<ImageView
android:layout_width="70dp"
android:layout_height="50dp"
android:layout_weight="2"
android:paddingLeft="10dp"
android:src="@drawable/audio" />
<TextView
android:id="@+id/scan_audio"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="5"
android:paddingLeft="10dp"
android:paddingRight="10dp"
android:text="Сканировать аудио" />
android:textSize="20dp"
<CheckBox
android:id="@+id/checkbox_audio"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="1"
android:paddingRight="10dp" />
</LinearLayout>
<LinearLayout
android:id="@+id/layout_video"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
```

А ҚОСЫМШАСЫ

```
        android:gravity="center"
        android:orientation="horizontal"
        android:padding="10dip">
    <ImageView
        android:layout_width="70dp"
        android:layout_height="50dp"
        android:layout_weight="2"
        android:paddingLeft="10dp"
        android:src="@drawable/video" />
    <TextView
        android:id="@+id/scan_video"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="5"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:text="Сканировать"
        android:textSize="20dp"
        видео" />
    <CheckBox
        android:id="@+id/checkbox_video"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:paddingRight="10dp" />
</LinearLayout>
<LinearLayout
    android:id="@+id/layout_image"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center"
    android:orientation="horizontal"
    android:padding="10dip">
    <ImageView
        android:layout_width="70dp"
        android:layout_height="50dp"
        android:layout_weight="2"
        android:paddingLeft="10dp"
        android:src="@drawable/image" />
    <TextView android:id="@+id/scan_image"
```

А ҚОСЫМШАСЫ

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="5"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:text="Сканировать"
        android:textSize="20dp"
        <CheckBox
        android:id="@+id/checkbox_image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:paddingRight="10dp"
        </LinearLayout>
        <LinearLayout
        android:id="@+id/layout_docs"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center"
        android:orientation="horizontal"
        android:padding="10dip">
        <ImageView
        android:layout_width="70dp"
        android:layout_height="50dp"
        android
            :layout_weight="2"
        android:paddingLeft="10dp"
        android:src="@drawable/docs"
        <TextView
        android:id="@+id/scan_docs"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="5"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:text="Сканировать"
        android:textSize="20dp"
        <CheckBox
        android:id="@+id/checkbox_docs"
        android:layout_width="wrap_content"
```

рисунки"
/>

/>

/>

документы"
/>

А ҚОСЫМШАСЫ

```
android:layout_height="wrap_content"
android:layout_weight="1"
android:paddingRight="10dp" />
</LinearLayout>
<LinearLayout
android:id="@+id/layout_full_scan"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:gravity="center"
android:orientation="horizontal"
android:padding="10dip">
<ImageView
android:layout_width="70dp"
android:layout_height="50dp"
android:layout_weight="2"
android:paddingLeft="10dp"
android:src="@drawable/full_scan" />
<TextView
android:id="@+id/full_scan"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="5"
android:paddingLeft="10dp"
android:paddingRight="10dp"
android:text="Полное сканирование" />
android:textSize="20dp"
<CheckBox
android:id="@+id/checkbox_full_scan"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="1"
android:paddingRight="10dp" />
</LinearLayout>
<LinearLayout
android:id="@+id/layout_button"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_weight="0.4"
android:gravity="center"
android:orientation="horizontal">
<Button
```

```
android:id="@+id/button"  
android:layout_width="fill_parent"  
android:layout_height="fill_parent"  
android:text="Сканировать"  
android:textSize="20dp"  
</LinearLayout>  
</LinearLayou
```

/>

В ҚОСЫШАСЫ

```
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.LinearLayout;
import android.widget.RadioButton;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
private TextView audio,video,image,docs,full_scan;
private LinearLayout
layout_audio,layout_video,layout_image,layout_docs,layout_full_scan;
private CheckBox
checkbox_audio,checkbox_video,checkbox_image,checkbox_docs,checkbox_full_
scan;
private Button button;
private boolean isAudio,isVideo,isImage,isDocs,isFullScan;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
audio=(TextView)findViewById(R.id.scan_audio);
layout_audio=(LinearLayout)findViewById(R.id.layout_audio);
checkbox_audio=(CheckBox)findViewById(R.id.checkbox_audio);
video=(TextView)findViewById(R.id.scan_video);
layout_video=(LinearLayout)findViewById(R.id.layout_video);
checkbox_video=(CheckBox)findViewById(R.id.checkbox_video);
image=(TextView)findViewById(R.id.scan_image);
layout_image=(LinearLayout)findViewById(R.id.layout_image);
checkbox_image=(CheckBox)findViewById(R.id.checkbox_image);

docs=(TextView)findViewById(R.id.scan_docs);
layout_docs=(LinearLayout)findViewById(R.id.layout_docs);
checkbox_docs=(CheckBox)findViewById(R.id.checkbox_docs);
full_scan=(TextView)findViewById(R.id.full_scan);
layout_full_scan=(LinearLayout)findViewById(R.id.layout_full_scan);
checkbox_full_scan=(CheckBox)findViewById(R.id.checkbox_full_scan);
```

В ҚОСЫШАСЫ

```
button=(Button)findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v) {
        if(checkbox_audio.isChecked()||checkbox_video.isChecked()||checkbox_image.isC
            hecked()||checkbox_docs.isChecked()||checkbox_full_scan.isChecked()) {
            Intent intent = new Intent(MainActivity.this,
                SearchActivity.class);
            intent.putExtra("isAudio", checkbox_audio.isChecked());
            intent.putExtra("isVideo", checkbox_video.isChecked());
            intent.putExtra("isImage", checkbox_image.isChecked());
            intent.putExtra("isDocs", checkbox_docs.isChecked());
            intent.putExtra("isFullScan", checkbox_full_scan.isChecked());
            startActivity(intent);
        }
    }
});
layout_audio.setOnClickListener(this);
layout_video.setOnClickListener(this);
layout_image.setOnClickListener(this);
layout_docs.setOnClickListener(this);
layout_full_scan.setOnClickListener(this);
//select checkbox_audio, unselect others
checkbox_audio.setOnCheckedChangeListener(new
    CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean isChecked)
        {
            if (isChecked)
            {
                checkbox_video.setChecked(false);
                checkbox_image.setChecked(false);
                checkbox_docs.setChecked(false);
                checkbox_full_scan.setChecked(false);
            }
        }
    });
//select checkbox_video, unselect others
checkbox_video.setOnCheckedChangeListener(new
    CompoundButton.OnCheckedChangeListener() {
        @Override
```

В ҚОСЫШАСЫ

```
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked)
{
    if (isChecked)
    {
        checkbox_audio.setChecked(false);
        checkbox_image.setChecked(false);
        checkbox_docs.setChecked(false);
        checkbox_full_scan.setChecked(false);
    }
}

//select checkbox_video, unselect others
checkbox_image.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener()
@Override
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked)
{
    if (isChecked)
    {
        checkbox_audio.setChecked(false);
        checkbox_video.setChecked(false);
        checkbox_docs.setChecked(false);
        checkbox_full_scan.setChecked(false);
    }
}

//select checkbox_docs, unselect others
checkbox_docs.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener()
@Override
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked)
{
    if (isChecked)
    {
        checkbox_audio.setChecked(false);
        checkbox_video.setChecked(false);
        checkbox_image.setChecked(false);
        checkbox_full_scan.setChecked(false);
    }
}
```

```
});
```

В ҚОСЫШАСЫ

```
//select checkbox_full_scan, unselect others
checkbox_full_scan.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
@Override
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked)
{
if (isChecked)
{
checkbox_audio.setChecked(false);
checkbox_video.setChecked(false);
checkbox_image.setChecked(false);
checkbox_docs.setChecked(false);
}
}
});
}
public void onClick(View v) {
isAudio = checkbox_audio.isChecked();
isVideo = checkbox_video.isChecked();
isImage = checkbox_image.isChecked();
isDocs = checkbox_docs.isChecked();
isFullScan = checkbox_full_scan.isChecked();
switch (v.getId())
{
case (R.id.layout_audio):
{
checkbox_audio.setChecked(!checkbox_audio.isChecked());
checkbox_video.setChecked(false);
checkbox_image.setChecked(false);
checkbox_docs.setChecked(false);
checkbox_full_scan.setChecked(false);

break;
}
case (R.id.layout_video):
{
checkbox_video.setChecked(false);
checkbox_video.setChecked(!checkbox_video.isChecked());
checkbox_image.setChecked(false);
checkbox_docs.setChecked(false);
checkbox_full_scan.setChecked(false);
}
}
}
```

```
checkbox_full_scan.setChecked(false);
break;
```

В ҚОСЫШАСЫ

```
case (R.id.layout_image):
{
checkbox_audio.setChecked(false);
checkbox_video.setChecked(false);
checkbox_image.setChecked(!checkbox_image.isChecked());
checkbox_docs.setChecked(false);
checkbox_full_scan.setChecked(false);
break;
}
case (R.id.layout_docs):
{
checkbox_audio.setChecked(false);
checkbox_video.setChecked(false);
checkbox_image.setChecked(false);
checkbox_docs.setChecked(!checkbox_docs.isChecked());
checkbox_full_scan.setChecked(false);
break;
}
case (R.id.layout_full_scan):
{
checkbox_audio.setChecked(false);
checkbox_video.setChecked(false);
checkbox_image.setChecked(false);
checkbox_docs.setChecked(false);
checkbox_full_scan.setChecked(!checkbox_full_scan.isChecked());
break;
}
}}
};
```

```
import android.content.DialogInterface;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.DefaultItemAnimator;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
```

```

import android.util.Log;
import android.view.View;
import android.webkit.MimeTypeMap;
В ҚОСЫШАСЫ

import android.widget.Button;
import android.widget.LinearLayout;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
public class SearchActivity extends AppCompatActivity {
    private final String TAG = SearchActivity.class.getSimpleName();
    private RecyclerView recyclerView;
    private TitleGroupAdapter mAdapter;
    private List<Object> groupTitleList = new ArrayList<>();
    private LinearLayout linlaHeaderProgress;
    private Button deleteButton;
    MessageDigest messageDigest;
    Map<String, List<String>> lists = new HashMap<>();
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.content_main);
        linlaHeaderProgress = (LinearLayout)
findViewById(R.id.linlaHeaderProgress);
        recyclerView = (RecyclerView) findViewById(R.id.recycler_view);
        mAdapter = new TitleGroupAdapter(SearchActivity.this);
        RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(this);
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setItemAnimator(new DefaultItemAnimator());
        recyclerView.setAdapter(mAdapter);
        // delete button

        deleteButton = (Button) findViewById(R.id.button_delete);
        deleteButton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {

```



```

        //alert dialog
        if(!mAdapter.getSelectedItems().isEmpty()) {
В ҚОСЫШАСЫ
        AlertDialog.Builder builder = new AlertDialog.Builder(SearchActivity.this);
            builder.setTitle("УДАЛИТЬ?");
                builder.setPositiveButton(android.R.string.ok, new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        //do something
                        for (int i = 0; i < mAdapter.getSelectedItems().size(); i++) {
                            //Log.e(TAG, "onClick: " + ((Title)
mAdapter.getSelectedItems().get(i)).getLink());
                                String linkDeleted = ((Title)
mAdapter.getSelectedItems().get(i)).getLink();
                                    List<Object> tg = mAdapter.getTitleGroupFeed();
                                        for (int j = 0; j < tg.size(); j++) {
                                            if (tg.get(j) instanceof Title) {
                                                Log.e(TAG, "selected: " + ((Title) tg.get(j)).getLink());
                                                if (linkDeleted.equals(((Title) tg.get(j)).getLink())) {
                                                    tg.remove(j);
                                                }
                                            }
                                        }
                                    }
                                File file = new File(linkDeleted);
                                    if (file.exists()) {
                                        file.delete();
                                    }
                                }
                            mAdapter.notifyDataSetChanged();
                                dialog.cancel();
                            }
                        });
                    builder.setNegativeButton(android.R.string.cancel, new
DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            dialog.cancel();
                        }
                    });
                AlertDialog dialog = builder.create();
                dialog.show();
                //end of alert dialog
            }

```

```
    }  
В ҚОСЫШАСЫ
```

```
});
```

```
    // progress bar  
    linlaHeaderProgress.setVisibility(View.VISIBLE);  
    try {  
        messageDigest = MessageDigest.getInstance("SHA-256");  
    } catch (NoSuchAlgorithmException e) {  
        e.printStackTrace();  
    }  
    final Handler handler = new Handler();  
    new Thread(new Runnable() {  
        @Override  
        public void run() {  
            startSearchDuplicateFiles();  
        }  
    }).start();  
}  
public void startSearchDuplicateFiles() {  
    //search duplicate files  
    File dir = Environment.getExternalStorageDirectory();  
    findDuplicateFiles(lists, dir);  
    //show duplicate files  
    boolean isAudio = getIntent().getExtras().getBoolean("isAudio");  
    boolean isVideo = getIntent().getExtras().getBoolean("isVideo");  
    boolean isImage = getIntent().getExtras().getBoolean("isImage");  
    boolean isDocs = getIntent().getExtras().getBoolean("isDocs");  
    boolean isFullScan = getIntent().getExtras().getBoolean("isFullScan");  
    int iter = 1;  
    for (List<String> list : lists.values()) {  
        if (list.size() > 1) {  
            String type_of_file = "";  
            for (String file : list) {  
                type_of_file = getMimeType(file);  
            }  
            if (type_of_file != null) {  
                //checking for audio  
                if (type_of_file.contains("audio") && isAudio) {  
                    //Log.e(TAG, "-----");  
                    Group group = new Group(iter);  
                    groupTitleList.add(group);  
                    mAdapter.setTitleGroupFeed(groupTitleList);  
                    for (String file : list) {
```

В ҚОСЫШАСЫ

```
String stitle = file.substring(file.lastIndexOf('/') + 1);
    //Log.e(TAG, "onCreate: " + getMimeType(file) + stitle);
    Title t = new Title(stitle, file);
    groupTitleList.add(t);
    mAdapter.setTitleGroupFeed(groupTitleList);
}
iter++;

}
//checking for video
if (type_of_file.contains("video") && isVideo) {
    //Log.e(TAG, "-----");
    Group group = new Group(iter);
    groupTitleList.add(group);
    mAdapter.setTitleGroupFeed(groupTitleList);
    for (String file : list) {
        String stitle = file.substring(file.lastIndexOf('/') + 1);
        //Log.e(TAG, "onCreate: " + getMimeType(file) + stitle);
        Title t = new Title(stitle, file);
        groupTitleList.add(t);
        mAdapter.setTitleGroupFeed(groupTitleList);
    }
    iter++;
}
//checking for image
if (type_of_file.contains("image") && isImage) {
    //Log.e(TAG, "-----");
    Group group = new Group(iter);
    groupTitleList.add(group);
    mAdapter.setTitleGroupFeed(groupTitleList);
    for (String file : list) {
        String stitle = file.substring(file.lastIndexOf('/') + 1);
        //Log.e(TAG, "onCreate: " + getMimeType(file) + stitle);
        Title t = new Title(stitle, file);
        groupTitleList.add(t);
        mAdapter.setTitleGroupFeed(groupTitleList);
    }
    iter++;
}
//checking for docs
if ((type_of_file.contains("text") ||
type_of_file.contains("application")) && isDocs) {
```

В ҚОСЫШАСЫ

```

//Log.e(TAG, "-----");
    Group group = new Group(iter);
    groupTitleList.add(group);
    mAdapter.setTitleGroupFeed(groupTitleList);
    for (String file : list) {
        String stitle = file.substring(file.lastIndexOf('/') + 1);
        //Log.e(TAG, "onCreate: " + getMimeType(file) + stitle);
        Title t = new Title(stitle, file);
        groupTitleList.add(t);
        mAdapter.setTitleGroupFeed(groupTitleList);
    }
    iter++;
}

if (isFullScan) {
    //checking for full scan
    //Log.e(TAG, "-----");
    Group group = new Group(iter);
    groupTitleList.add(group);
    mAdapter.setTitleGroupFeed(groupTitleList);
    for (String file : list) {
        String stitle = file.substring(file.lastIndexOf('/') + 1);
        //Log.e(TAG, "onCreate: " + getMimeType(file) + stitle);
        Title t = new Title(stitle, file);
        groupTitleList.add(t);
        mAdapter.setTitleGroupFeed(groupTitleList);
    }
    iter++;
}

}

}

runOnUiThread(new Runnable() {
    @Override
    public void run() {
        mAdapter.notifyDataSetChanged();
        linlaHeaderProgress.setVisibility(View.GONE);
    }
});
}

public static String getMimeType(String url) {
    String type = null;
    В КОСЫХАКЫ
    String extension = MimeTypeMap.getFileExtensionFromUrl(url);

```

```

        if (extension != null) {
            MimeTypesMap type = MimeTypesMap.getSingleton().getMimeTypeFromExtension(extension);
        }
        return type;
    }
    public void findDuplicateFiles(Map<String, List<String>> filesList, File
directory) {
        for (File dirChild : directory.listFiles()) {
            if (dirChild.isDirectory()) {
                findDuplicateFiles(filesList, dirChild);
            } else {
                try {
                    FileInputStream fileInput = new FileInputStream(dirChild);
                    if (dirChild.length() > 1073741000)
                        return;
                    byte fileData[] = new byte[(int) dirChild.length()];
                    fileInput.read(fileData);
                    fileInput.close();
                    String uniqueFileHash = new BigInteger(1,
messageDigest.digest(fileData)).toString(16);

                    List<String> identicalList = filesList.get(uniqueFileHash);
                    if (identicalList == null) {
                        identicalList = new LinkedList<>();
                    }
                    identicalList.add(dirChild.getAbsolutePath());
                    filesList.put(uniqueFileHash, identicalList);
                } catch (IOException e) {
                    throw new RuntimeException("cannot read file " +
dirChild.getAbsolutePath(), e);
                }
            }
        }
    }
}

```

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ғылыми жетекшінің пікірі

Дипломдық жұмыс

Құлманов Дәулетхан Полатұлы

5В070300 – Ақпараттық жүйелер

Тақырыбы: Қайталанатын файлдарды табуға арналған мобильді қосымша жасау

Бұл дипломдық жұмыс өзінің логикалық құрылымымен ерекшеленген. Түсіндірме жобаның құрамы кіріспеден, 3 бөлімнен, қорытындыдан, әдебиеттер тізімінен және қосымшадан тұрады.

Менің пікірімше, диплом жобалаушы алдына қойылған тапсырманы толығымен орындады және кейінгі технологияларын меңгергендігін көрсетті.

Жалпы дипломдық жоба профессионалдық деңгейде орындалған. Түсіндірме жазба сауатты бейнеленген, жоба бойынша барлық қажетті ақпараттар бар.

Кемшілік ретінде кейбір шағын стилистикалық қателерді атап кетуге болады.

Жоғарыда айтылғандарға байланысты, дипломдық жұмыс 5В070300 – «Ақпараттық жүйелер» мамандығының бітіру жұмыстарына қойылатын талаптарына сәйкес және дипломдық жұмыс қорғауға жіберіле алады, ал оның авторы Құлманов Дәулетхан Полатұлы бакалавр академиялық дәрежесін алуға лайықты деп есептеймін.

Ғылыми жетекші

Лектор



Зиро А.А.

«24» мая 2021 ж.

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Құлманов Дәулетхан Полатұлы

Название: Қайталанатын файлдарды табуға арналған мобильді қосымша жасау

Координатор: Зиро А.А.

Коэффициент подобия 1: 9,74

Коэффициент подобия 2: 5,65

Замена букв: 24

Интервалы: 0

Микропробелы: 0

Белые знаки: 0

После анализа Отчета подобия констатирую следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

После анализа отчета по плагиату и работы дипломника выявлено, что заимствования являются добросовестными и не обладают признаками плагиата, так в основном связаны с применением общеизвестных терминов.

...25.05.2021.....

Дата

.....

Подпись Научного руководителя

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Құлманов Дәулетхан Полатұлы

Название: Қайталанатын файлдарды табуға арналған мобильді қосымша жасау

Координатор: Зиро А.А.

Коэффициент подобия 1: 9,74

Коэффициент подобия 2: 5,65

Замена букв: 24

Интервалы: 0

Микропробелы: 0

Белые знаки: 0

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....
.....
.....
.....

27.05.2021



Дата

Подпись заведующего кафедрой /

начальника структурного подразделения