

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт Информационных систем и геотехнологий
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

БАКАЛАВРСКАЯ РАБОТА

На тему Проектирование геоинформационной системы

Исполнитель Антонов Алексей Владимирович
(фамилия, имя, отчество)
Руководитель доктор географических наук, профессор
(ученая степень, ученое звание)
Шилин Михаил Борисович
(фамилия, имя, отчество)

«К защите допускаю»
Заведующий кафедрой _____
(подпись)

(ученая степень, ученое звание)

(фамилия, имя, отчество)
«__» _____ 20__ г.

Санкт-Петербург
2022

Список сокращений и специальных терминов

- ВКР – Выпускная квалификационная работа
- ВРМН – (Business Process Model and Notation) нотация и модель бизнес-процессов
- ИА – (Information availability) доступность информации
- ИС – Информационная система
- ИТ – Информационная технология
- UML – (Unified Modeling Language) унифицированный язык моделирования
- ООП – Объектно-ориентированное проектирование
- ПО – Программное обеспечение
- Java – строго типизированный объектно-ориентированный язык программирования
- СУБД – Система управления базой данных
- NoSQL – обозначение широкого класса разнородных систем управления базами данных, существенно отличающихся от традиционных реляционных СУБД с доступом к данным средствами языка SQL.
- API – (Application Programming Interface) программный интерфейс приложения
- DRP – (Disaster Recovery Plan) План аварийного восстановления

Оглавление

Список сокращений и специальных терминов.....	2
ВВЕДЕНИЕ	4
ГЛАВА 1. ПРЕДПРОЕКТНЫЙ АНАЛИЗ.	6
1.1 Анализ предметной области	6
1.2 Выделение необходимого функционала	11
1.3 Выбор средств разработки и их обоснование	12
1.4 Сравнительный анализ аналогов ИС	13
1.5 Анализ объекта исследования	18
ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИС.	23
2.1 Концептуальное проектирование	23
2.2 Функциональное проектирование.	24
2.3 Объектно-ориентированное проектирование.	26
2.4 Диаграмма вариантов использования (USE-CASE).	27
2.5 Диаграммы последовательности.	28
2.6 Диаграмма классов.	30
2.7 Диаграмма развёртывания.	32
ГЛАВА 3 РАЗРАБОТКА ИС	35
3.1 Основные элементы клиентского интерфейса.	35
3.2 Технологические процессы, обрабатывающие данные.	38
3.3 Схема восстановления и сохранения информации.	39
3.4 Экономическая эффективность	43
3.5 Сроки окупаемости.	46
ЗАКЛЮЧЕНИЕ	48
Список использованных источников и литературы	49
ПРИЛОЖЕНИЕ А	51
ПРИЛОЖЕНИЕ Б	54
ПРИЛОЖЕНИЕ В	56

ВВЕДЕНИЕ

В течении многих веков в истории человечества присутствуют вооруженные конфликты. Баталии развиваются, становятся более масштабными и ожесточенными. Средства уничтожения становятся более универсальными и смертоносными. Меняется мир, и война меняется вместе с ним. Эти громкие и зловещие события меняют облик государств, и будут существовать, пока последние не перестанут иметь видимых границ. [6, с. 49]

Среди определенных общественных групп, проявляющих интерес к истории в современном мире, наблюдается интерес к военной технике. Отдельно следует обратить внимание на категорию бронированных боевых машин, обозначенные коротким словом «танк». Исходя из вышесказанного, следует углубиться в тему и понять, по каким причинам данный вид войск стал настолько популярен.

Для человека, интересующегося как инженерией, так и историей, интересно наблюдать за быстрым развитием танков от примитивных ромбовидных форм Первой мировой войны до современных основных боевых танков, особенно если обращать внимание на тупиковые и перспективные пути развития. Некоторое время в 1930-х годах преобладало мнение, что будущее за многобашенными танками, однако, в результате многочисленных испытаний большого круга стран, от этой идеи отказались. Снаряды с фугасным воздействием делали броню устаревшей до тех пор, пока технический прогресс опроверг данную теорию, что сильно отразилось на конструкторах. История танка — это история постоянных экспериментов и разработок, которые в настоящее время считаются вершиной конструкции танков с основных боевых танков текущего поколения, но даже сейчас эксперименты не прекращаются [5, с. 49].

С появлением информационных технологий всю информацию о бронированных боевых машинах стало значительно легче почерпнуть. Однако, информационные источники по данному типу техники могут быть довольно сильно перегружены не нужной базовому пользователю информацией.

Актуальность системы заключается в довольно большом интересе боевой бронетехникой в современном мире. Человек, который проявляет интерес к данному типу боевой техники, приобщается к истории, коллекционирует модели, ходит в музеи и развивает патриотические настроения. Необходимо также помнить о большой популярности танковой тематики в нашей стране и о большом росте популярности такого рода информационных систем, т. к. в большинстве случаев информация преподносится без конкретики, что усложняет поиск.

Объектом исследования выступает область военной истории, в большей степени связанной с бронированными боевыми машинами.

Предметом исследования выступает проектирование приложения с элементами геоинформационной системы для удобного отображения исторической информации о гусеничной бронетехнике в различных временных рамках.

Целью выпускной квалификационной работы является проектирование геоинформационной системы по ознакомлению с крупной бронетехникой с целью упрощения поиска, решения проблемы разрозненности информации в области получения информации по бронированным боевым машинам и восприятия информации заинтересованными пользователями. Проведение исследования и анализ выбранной области для формулировки конкретных требований, предъявляемых к информационной системе.

Задачи выпускной квалификационной работы:

- Провести анализ информационной системы и выбранной области с выделением требований;
- Провести анализ типовых проектных решений с целью использования в собственных проектах, а также подобрать необходимые инструменты для дальнейшей разработки;
- Спроектировать информационную систему с учетом проведённого комплексного анализа, с дальнейшей разработкой прототипа;
- Сделать выводы по проделанной работе и провести анализ дальнейших перспектив развития системы.

Инструментарий проведения работы.

В ходе выполнения практической части ВКР, будут использованы следующие вспомогательные программы: облачная база данных FirebaseDatabase, интегрированная среда разработки Android Studio.

ГЛАВА 1. ПРЕДПРОЕКТНЫЙ АНАЛИЗ.

1.1 Анализ предметной области

История танка начинается с Первой мировой войны, когда в качестве ответа на проблемы позиционной войны были представлены боевые бронированные вездеходы, открывшие новую эру механизированной войны [3, с. 49] (Рисунок 1).

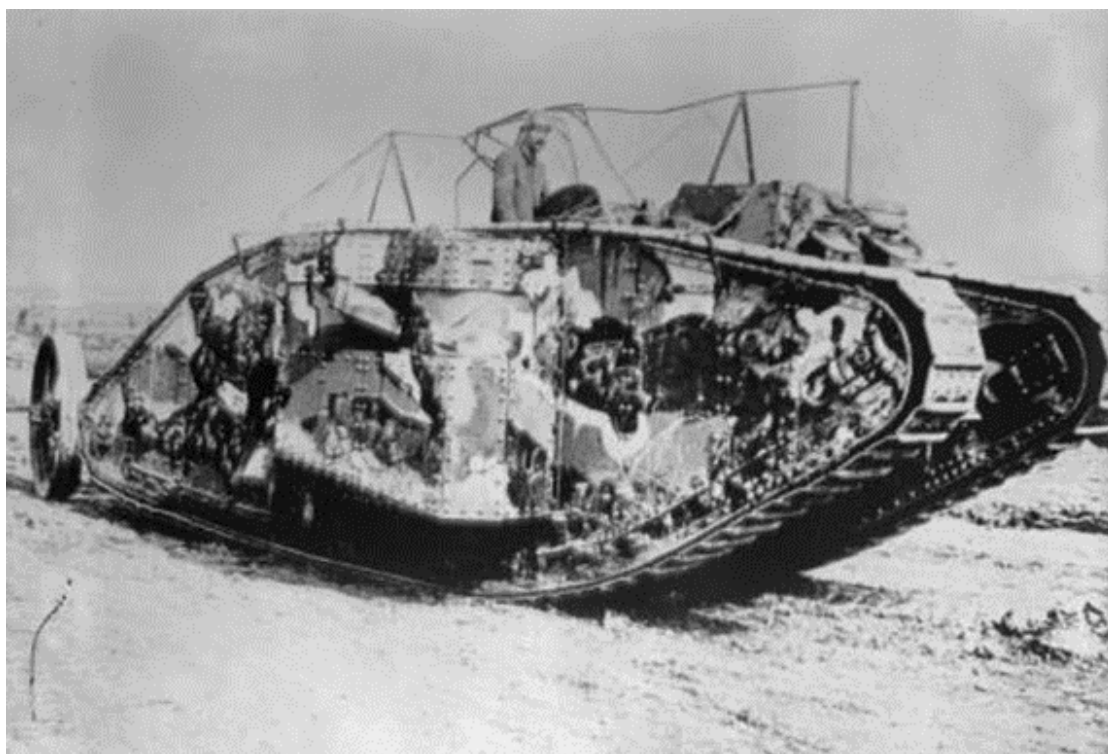


Рисунок 1 — танк Первой мировой войны Mark I [3, с.49]

Первая официальная фотография танка в бою, сделанная в битве при Флер-Курселетт 15 сентября 1916 года. Это танк Mark I с рулевым оперением в задней части машины, которое исчезло во многих более поздних моделях. Первые танки были медленными и ненадежными, о чем свидетельствует тот факт, что из 49 танков, задействованных в бою, только 25 фактически двинулись вперед в начале атаки. [3, с.49]

К началу Второй мировой войны конструкция танков значительно продвинулась, и танки в большом количестве использовались на всех сухопутных театрах военных действий. Этот временной промежуток стал началом триумфа мирового танкостроения. Никогда в истории не было произведено и отправлено на боевые действия столь огромное количество танков по всему миру.

Знаковые танковые битвы Второй мировой войны:

Битва при Бродах (1941)

Ожесточенное танковое сражение между немецкой и советской армией. Оно на протяжении 2 лет оставалось крупнейшим танковым сражением Второй мировой войны до Курской битвы. Несмотря на численное превосходство советских войск, немецкие силы одержали победу. В битве участвовало 3300 боевых машин. (Рисунок 2)



Рисунок 2 — Битва при Бродах [20, с 49]

Битва за Прохоровку (1943)

Данное танковое сражение Второй мировой войны было частью Курской битвы и является одним из крупнейших танковых сражений в военной истории. Несмотря на то, что победа в этой битве не была сразу очевидна, битва продемонстрировала неизменную силу и решимость советских солдат (Рисунок 3).



Рисунок 3 — Битва за Прохоровку [20, с 49]

Вторая битва при Эль-Аламейне (1942)

Немецкие войска проиграли это сражение и не смогли овладеть Египтом и Суэцким каналом. В победе Союзников танки сыграли решающую роль (Рисунок 4).



Рисунок 4 — Вторая битва при Эль-Аламейне [20, с 49]

Танки преобладали в большом количестве, где можно было наблюдать солдат. Они участвовали в большинстве сражений Второй мировой войны, некоторые из них стали легендарными, как Курская битва, одно из крупнейших танковых столкновений в истории человечества. [1, с.49]

Во время холодной войны (1946–1990) советское господство в Варшавском договоре привело к эффективной стандартизации нескольких конструкций танков. Для примера, Франция, Германия, США и Великобритания ранее разрабатывали собственные конструкции танков, но позднее была предпринята попытка стандартизировать свои конструкции, в то время как более мелкие страны НАТО закупали или адаптировали данные конструкции, что способствовало появлению основных боевых танков общего назначения. Несмотря на изменения в современной войне, планов по избавлению от танков пока не предусмотрено. Танк по-прежнему является основой наземных боевых действий в 21 веке (Рисунок 5). [1, с.49]



Рисунок 5 — Пример основного боевого танка (ОБТ)

Основная проблема

При работе с данными о боевой технике есть вероятность столкнуться с проблемой разрозненности информации. Часть документации на боевые машины либо были утеряны со временем, либо до сих пор пылятся в архивах в ожидании своего использования. Также стоит упомянуть, что в большом количестве современных информационных преобладает разрозненная информация. Когда человек пытается найти информации по боевой гусеничной технике, ему придется столкнуться с тем, что полную и точную информацию в одном конкретном источнике найти не удастся. Ему придется посещать несколько источников.

Преимущество визуализации информации

Включительно, можно подчеркнуть визуализацию данных. Как правило, человеческий мозг запоминает оптическую информацию значительно быстрее, чем при изучении текстов, что позволяет воссоздать в голове картину

полученного материала. При этом следует помнить, что вся визуальная информация должна быть дополнена корректной текстовой информацией, чтобы у пользователя вырисовывалась четкое представление о технике.

Дизайн сайтов и приложений по военной тематике

Красивым оформлением и дизайном могут отличаться, преимущественно коммерческие компании. Базы знаний таких игровых проектов как War Thunder (Рисунок 6) предоставляют информацию в удобном формате, при этом ориентируясь на архивные документы. Источники, созданные для более исторического ознакомления, в качестве дополнительного примера можно представить энциклопедию Airwar.ru, напротив, предлагают плохой и очень неудобный дизайн (Рисунок 7) [8, с. 49].

Как было упомянуто выше, созданные проекты созданы для общего пользования единицами и крупными фирмами для получения прибыли.



Рисунок 6 — База знаний игрового проекта «War Thunder»



Рисунок 7 — Большая авиационная энциклопедия Airwar.ru

На основе вышесказанного можно приступить к первым этапам создания приложения. На первых стадиях проектирования приложения важным является анализ предметной области, т.е. определение объектов предметной области и связи между ними.

Информационная система «Tanks on the map» является Android-приложением, предназначенным для корректного и удобного предоставления информации о гусеничной технике при помощи карты.

В приложении необходимо реализовать добавление, редактирование и просмотр данных о военной технике. Целевой аудиторией приложения будут являться группы лиц, проявляющие интерес к боевой технике со всего мира. Допустимо предположить, некоторые из круга индивидов, интересующихся данной темой знатоков истории примут решение получить информацию о интересующей конкретных личностей боевой машине, в таком случае придется потратить некоторое время на поиск все необходимого, что не всегда удобно.

1.2 Выделение необходимого функционала

Изначально, любая ИС, связанная с геоинформационными технологиями, позволяет создавать, редактировать, удалять и сохранять информацию о изображенных на карте объектах. Дополнительные же функции разнятся в зависимости от вида и направления работы. Так, для проектируемой системы характерны такие функции как:

- взаимодействие пользователей с картой;
- регистрация пользователя;
- создание/редактирование/удаление информации разработчиком;
- приём и обработка получаемой информации от пользователей до попадания в основную базу данных;

С другой стороны, пользователи системы управления тестированием ожидают увидеть следующее:

- удобная установка и поддержка;
- создание и управление информацией со стороны пользователя;

- создание пользователей и ролей пользователей;

Все вышеперечисленное при дальнейшей разработке будет учитываться (например, такие критерии как удобство, понятность и логичность системы).

1.3 Выбор средств разработки и их обоснование

Необходимым средством разработки являются CASE - средства (Computer Aided Software/System Engineering).

Под термином CASE понимаются понимаются программные средства, поддерживающие процессы создания и сопровождения информационных систем, включая анализ и формулировку требований, проектирование прикладного программного обеспечения и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы.

Так, среди них можно выделить:

- средства облачных баз данных (Firebase Database);
- средства разработки приложений (Android Studio);
- интерактивная карта (Google Maps).

Данные средства имеют определённые требования, например, минимальное аппаратное и программное обеспечение:

- – тип: мобильное устройство;
- – операционная система: Android версии 11;
- – оперативная память: 4 ГБ;
- – место на диске: 1 ГБ;
- – API версии 28 и выше.

Использование CASE-средств позволяют облегчить процесс дальнейшей разработки и структурировать его.

1.4 Сравнительный анализ аналогов ИС

При рассмотрении информационных ресурсов было выявлено множество сайтов и источников в области танковой истории. Сравнительный анализ подобных источников заключается в выделении сильных и слабых сторон разработок конкурирующих ресурсов и сравнение их функционала. Первым аналогом является сайт Military.com (Рисунок 8) [9, с. 49].

Документы показывают морские захоронения 13 моряков USS Indianapolis



На этой фотографии 10 июля 1945 года, предоставленной ВМС США, USS Indianapolis (CA 35) показан с военно-морской верфи острова Маре в Вальехо, штат Калифорния.

27 мая 2022 года
Associated Press / Одри Макэвой

ГОНОЛУЛУ — моряку 2-го класса Джорджу Дэвиду Пейну было всего 17 лет, когда японская подводная лодка торпедировала его корабль в последние недели Второй мировой войны, потопив судно и убив его

- Военное мнение
- Армия
- Военно-морской флот
- ВВС
- Корпус морской пехоты
- Береговая охрана
- Космические силы

Выберите сервис

Армия	Морские пехотинцы	Военно-морской флот	ВВС
Национальная гвардия	Береговая охрана	Космические силы	Супруг
Вход			

Самые популярные военные новости



Сокращение пособий на прожиточный минимум "затрудняет просто существование", - говорят семьи военных в Германии



Последний известный выживший в резне военнопленных Второй мировой войны Умер в 99 лет



Корпус морской пехоты рассматривает возможность отказа от

Рисунок 8 — Military.com

Слабая сторона сайта «Military» заключается в том, что он представляет из себя более информационный портал армии США, чем базу знаний по технике. На данном сайте можно найти рассказы ветеранов американской армии, статьи об армейской жизни, но при этом нет полной информации о танковых войсках. Ниже приведен пример изображенной техники, которую запечатлели представители прессы (Рисунок 9).



Рисунок 9 — Пример фотографии Military.com

Сильной чертой можно отметить большое разнообразие информации. Приведена подробная информация о технике армии США, состоящая на вооружении на данный момент. В дополнении к вышесказанному идут множественные новостные сводки, всплывающие окна и медиа файлы.

Далее стоит рассмотреть базу знаний MMO – проекта World of Tanks. Это коммерческий проект компании Wargaming.net представленный в виде компьютерной игры. Для данного игрового проекта была создана база знаний, представленная на рисунке 10.

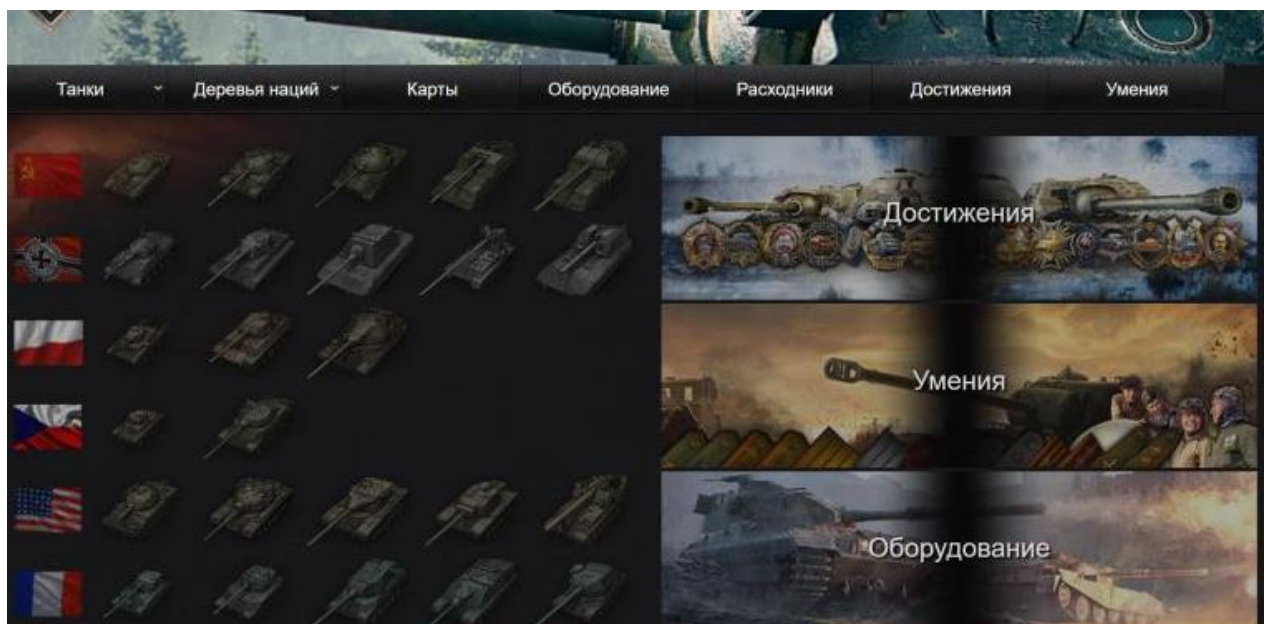


Рисунок 10 — База знаний World of Tanks Base

Сильной стороной WoTBase является огромное разнообразие техники и возможность использования приложения на Android. Также следует упомянуть о возможности просмотра техники в трехмерном пространстве.

Недостатком можно подчеркнуть узкую направленность контента, предназначенного для пользователей данного онлайн проекта. Ниже представлен 3D макет танка Т - 44 образца 1944 года, смоделированного для данной базы знаний (Рисунок 11).



Рисунок 11 —Модель танка Т 44 для базы знаний World of Tanks

Однако, для ознакомления с более подробной исторической информацией приходится обращаться к другим источникам. Например, с интернет-энциклопедией The Online Tank Museum (Рисунок 12)



Рисунок 12 — Интернет-энциклопедия The Online Tank Museum

Преимуществом данной энциклопедии можно отметить огромное количество изображения и фотографий. Огромный пласт информации с подробными чертежами и схемами. Также прикладываются подлинные документы конструкторов (Рисунок 13).

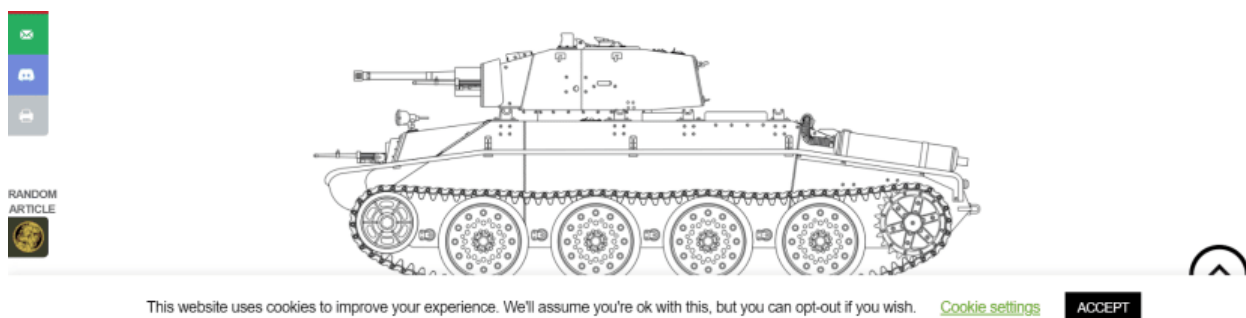


Рисунок 13 – Пример рисунка The Online Tank Museum

К недостатку можно отнести не самый совершенный интерфейс. За доступность пришлось заплатить большим количеством всплывающей рекламы

Далее приведена таблица аналогов исходя из функций, которыми должна обладать хорошее приложение (см. Таблица 1.1).

Таблица 1.1- Сравнение аналогов по функционалу

Функционал	World of Tanks Base	Military.com	The Online Tank Museum
Создание и ведение информации	+	+	+

технике/истории создания/документации			
Простота использования базовым пользователем	+	-	+
Авторизация	+	+	+
Карта	-	-	-
Редактирование	+	-	+

Таблица 1.2. – Возможные функции ИС.

Необходимые	Желательные	Возможные	Отсутствующие
Взаимодействие пользователей с картой	Просмотр и взаимодействие пользователей с интерактивной картой	Интеграция с системой управления версиями	Возможность распределения ролей в системе, кроме тех, что присутствуют в регистрации пользователя.
Регистрация пользователя	Создание простых отчётов	Персональная страница пользователя	Ранжирование пользователей в зависимости от метода использования
Авторизация пользователя	Возможность создания пользователем тем и сообщений на вкладках приложения	Возможность создания каналов чата	Возможность редактирования карты
Взаимодействие с остальными пользователями по средством службы поддержки	Передача файлов внутри чата	Поиск по сообщениям чата	Голосовой и видео чат
Получение программой доступа к геолокации пользователя	Поиск по записям и сообщениям (для модератора)	Закрепление сообщений в чате	Поиск и создание упоминаний в чате

Необходимые	Желательные	Возможные	Отсутствующие
Создание модератором постов(меток) на карте	Показатель активности и неактивности пользователей на карте	Создание настраиваемых пользователем оповещений	Создание приватного чата

По итогу сравнительного анализа можно сказать, что ориентированный на заинтересованного пользователя The Online Tank Museum в долгосрочной перспективе имеет множество вышесказанных преимуществ, но также обладает недостатком – средним дизайн. Остальные ресурсы имеют похожую функциональность, на которых проблематично найти информацию, не все удобны для базового пользователя. Со сравнительного анализа можно сделать вывод: выполнение данного проекта, а именно создание геоинформационного приложения может привлечь нужную аудиторию, учитывая найденные недостатки и опередив в этом плане аналогичные продукты.

На основе полученных данных следующим действием будет проведение анализа объекта исследования.

1.5 Анализ объекта исследования

Системный анализ — научный метод, который отличается междисциплинарным подходом к решению сложных проблем.

В рамках выпускной квалификационной работы были использованы следующие методологии анализа:

- SWOT;
- VCM;
- BRP;
- ISA.

Для определения необходимости разработки приложения был проведен SWOT-анализ.

SWOT-анализ (сильные и слабые стороны, возможности и угрозы) — это структура, используемая для оценки конкурентной позиции компании и разработки стратегического планирования. SWOT-анализ оценивает внутренние и внешние факторы, а также текущий и будущий потенциал.

Эта структура считается мощной поддержкой для принятия решений, поскольку она позволяет раскрывать возможности для достижения успеха, которые ранее не были сформулированы, и выявляет угрозы до того, как они станут чрезмерно обременительными.

Таблица 1.3 - SWOT-анализ

SWOT-анализ разрабатываемой системы	
S	<ol style="list-style-type: none"> 1. Простота использования. 2. Возможность просмотра информации на карте. 3. Отображение точек на карте в (с приложенными фото).
W	<ol style="list-style-type: none"> 1. Отсутствие финансирования. 2. Неопытность разработчика. 3. Низкая скорость обмена данных с сервером при нагрузке.
O	<ol style="list-style-type: none"> 1. Малое количество прямых аналогов системы. 2. Возможность дальнейшей доработки пользователями (открытый исходный код). 3. Бесплатное пользование.
T	<ol style="list-style-type: none"> 1. Появление аналогов. 2. Разработку могут доработать и продавать другие компании. 3. Сомнительная актуальность.

После проведения SWOT-анализа наглядно видны все минусы и плюсы разрабатываемого приложения. Дальнейшим действием является разбор подхода VCM.

Метод ценностных цепочек — VCM — позволяет оценить конкурентные преимущества с помощью анализа всей цепочки видов деятельности в организации, начиная от получения сырья до конечной продукции, продаваемой и поставляемой потребителям.

Модель служит цели уяснения того, какая конфигурация цепочки ценности сулит наибольшие конкурентные преимущества

1. Оценить информационную емкость продуктов и процессов.

Отображение техники на карте и информации о ней,

Размещение танков, исторических справок, характеристик.

2. Оценить роль ИТ в отраслевой структуре.

Экономия времени. Полезная информация.

3. Выявить и ранжировать способы, с помощью которых ИТ создает конкурентное преимущество.

Удобство (информация отображена на карте)

Доступность (бесплатное приложение)

4. Рассмотреть, каким образом ИТ может создать новое направление в бизнесе.

5. Разработать план, направленный на извлечение выгод от использования ИТ.

Поисковые сервисы, Нахождение информации, Разные виды техники.

BPR

Алгоритм информационной системы — это подход к планированию разработки ИС с использованием методов реинжиниринга бизнес-процессов (BPR-методов) основан на допущении, что современные организации должны реконструировать себя и отказаться от функциональной декомпозиции, иерархических структур и принципов приоритетности повседневных нужд, которые они сегодня используют. Ниже представлен алгоритм разрабатываемой системы (Рисунок 14).

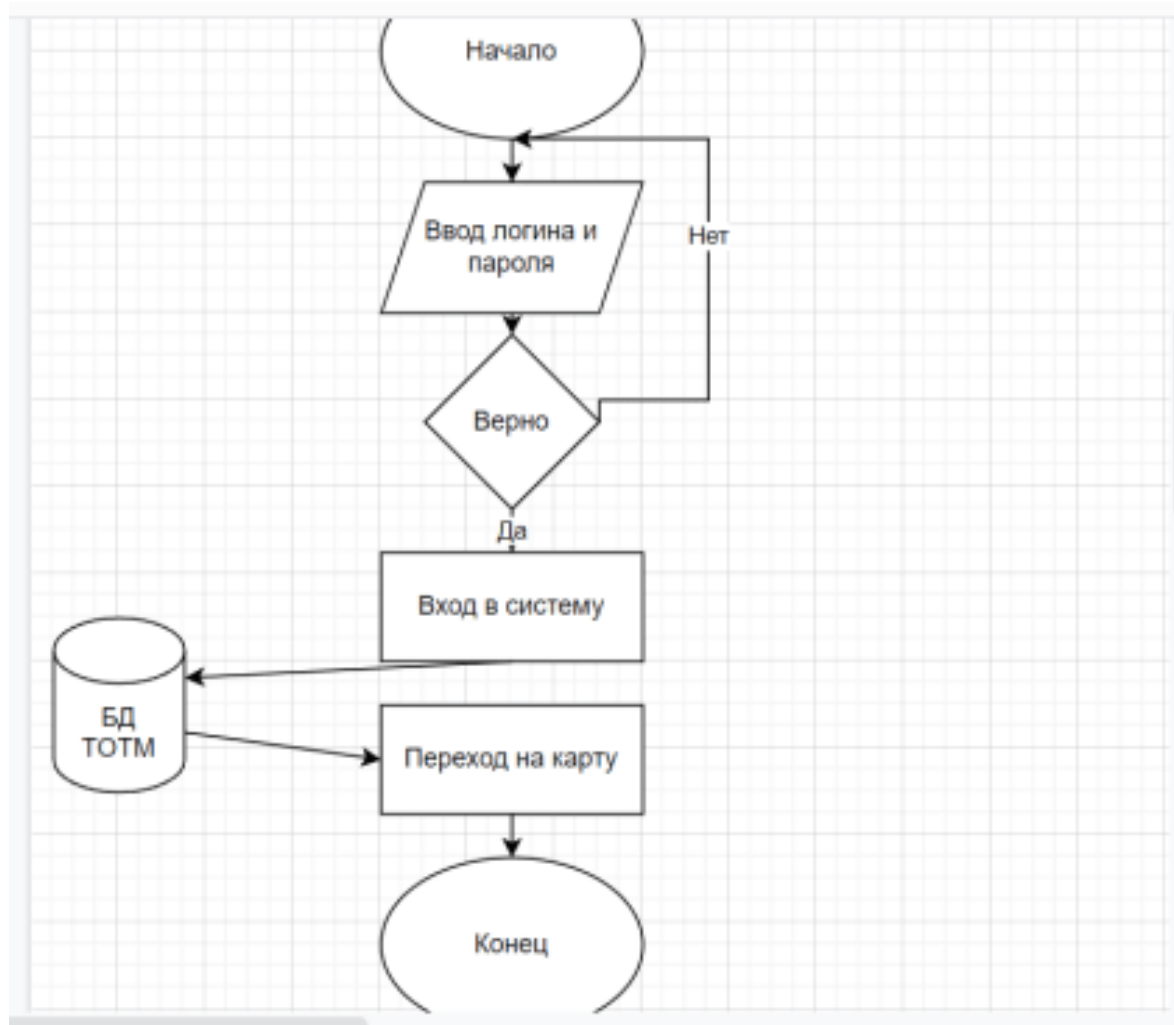


Рисунок 14 — Алгоритм системы «Tanks on the map»

Диаграмма Ганта

В самом начале XX века Генри Гант произвел революцию в сфере управления проектами, создав диаграммы Ганта. В то время все диаграммы составлялись на листах бумаги. По мере развития компьютеров в 1980-х годах диаграммы Ганта становились все более сложными и детализированными. На сегодняшний день они по-прежнему являются одним из наиболее популярных инструментов управления проектами.

Для определения сроков реализации проекта используется диаграмма Ганта (столбчатая диаграмма)[4, с.49], которая отображает процесс разработки в зависимости от поставленных сроков (Рисунок 15).

Диаграмма Ганта является одним из методов планирования проектов.

Наименование	Месяц															
	Сен	Окт	Ноя	Дек	Янв	Фев	Мар	Апр	Май	Июн	Июл	Авг	Сен	Окт	Ноя	Дек
Фаза анализа и планирования требований	■	■	■	■	■											
Фаза проектирования						■	■	■	■							
Фаза построения										■	■	■	■	■	■	
Фаза внедрения															■	■
Год	2020				2021											

Рисунок 15 — Диаграмма Ганта

ISA

Необходимо провести анализ по методологии ISA (Таблица 1.3.). Таким образом мы выясним роли и составляющие образ объекта проектирования в целом. Объектом проектирования выступает проектируемая геоинформационная система.

Таблица 1.4 - подход ISA

БД, Сервер БД характеристики приложения	Процесс регистрации, обработки информации (сохранение, изменение, удаление в БД)	Сервер БД	Администратор	В момент работы с БД	Полноценное функционирование с данными, их обработка и изменение
--	--	-----------	---------------	----------------------	--

Данные, информация о конкурентах	Контроль и анализ данных источников	Покупка или аренда хостинга, аренда сервера	Администратор	Моменты обеспечения ИС	Выведении ИС на рынок, помощь людям в поиске информации
Информация о машинах и их характеристиках из БД, характеристики приложения	Формирование данных в БД о технике	БД, хостинг, Сервер БД	Администратор	В момент добавления новой техники	Стабильность работы системы

Продолжение таблицы 1.4 – ISA-анализ приложения «Tanks on the map»

Юридические документы	Оформление документов	Лицензии	Проверяющие инстанции	При запросе нормативных документов	Юридическая грамотности
Боевые машины	Количество добавленных и найденных танков	Офис разработчика	Администратор	Передача данных в ИС	Расширение БД о танках, увеличение вероятности нахождения предметов

ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИС.

Проектирование систем — это процесс определения элементов системы, таких как модули, архитектура, компоненты и их интерфейсы и данные для системы, на основе заданных требований[4, с.49].

Разные люди используют слово «проектирование» в разных контекстах. Когда ИТ-специалисты говорят о проектировании, они имеют в виду бизнес-процессы. Прежде чем разрабатывать, разрабатывать и внедрять решения, необходимо проанализировать проблемы и задокументировать требования.

Далее приведены модели системы или её основных частей в различных методах проектирования.

2.1 Концептуальное проектирование

Многие планы проектов теперь объединяют этапы концептуального и схематического проектирования, используя термин «концепция» для описания этого этапа разработки проекта.

Процесс создания концепции, т.е. концептуальное проектирование, должен являться исходным отправным пунктом всего процесса проектирования. Концепция представляет собой обобщенное структурное отображение системы и укрупненную взаимосвязь основных ее блоков, каждый из которых представлен на уровне "черного ящика".

Следует заметить, что понятие концептуального проектирования может быть применимо не только к техническим разработкам, но и к выработке подходов в решении задач управленческих, финансовых, информационных и других видов деятельности.

Концепция не является нормативным документом и не играет роль нормативно правового акта. Сам процесс построения концепции в наименьшей мере поддается формализации и в основном базируется на эвристических процедурах. Концепция разрабатывается на основе системно-логического анализа на уровне содержательного рассмотрения назначения системы, специфических особенностей ее основных конструктивных элементов и системных взаимосвязей между ними.

Благодаря концептуальной модели, можно выяснить связи и взаимоотношение между объектами системы, что поможет при дальнейшей разработке. Для создания концептуальной модели ИС используется схематическое представление системы, которое представлено на рисунке 16.

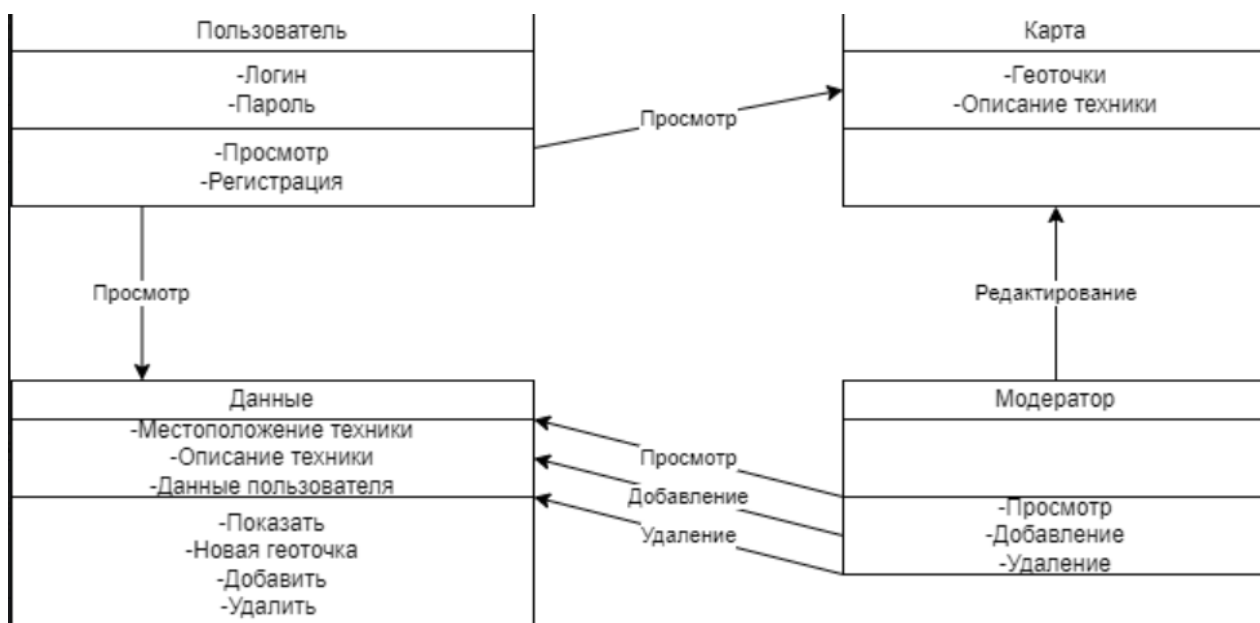


Рисунок 16 — Схематическое представление системы

2.2 Функциональное проектирование.

Функциональное проектирование — это парадигма, используемая для упрощения проектирования аппаратных и программных устройств, таких как компьютерное программное обеспечение и трехмерные модели. Функциональное проектирование гарантирует, что каждая модульная часть устройства несет только одну ответственность и выполняет эту функцию с минимальными побочными эффектами для других частей. Функционально разработанные модули, как правило, имеют низкую связанность.

Системы с функционально спроектированными частями легче модифицировать, потому что каждая часть делает только то, для чего заявлена.

Поскольку техническое обслуживание составляет более 3/4 срока службы успешной системы, эта функция является решающим преимуществом. Это также упрощает понимание и документирование системы, что упрощает обучение. В результате практический срок службы функциональной системы увеличивается.

Для построения функциональной модели системы использовались диаграммы BPMN или же система условных обозначений и их описания для моделирования бизнес-процессов. Готовые текущие схемы разрабатываемой ИС отмечены на рисунках 17,18 и 19.

BPMN-схемы наглядно и подробно демонстрируют последовательность рабочих действий и перемещение информационных потоков, необходимых для выполнения процесса, а потому являются одним из ключевых инструментов для создания хороших функциональных моделей.

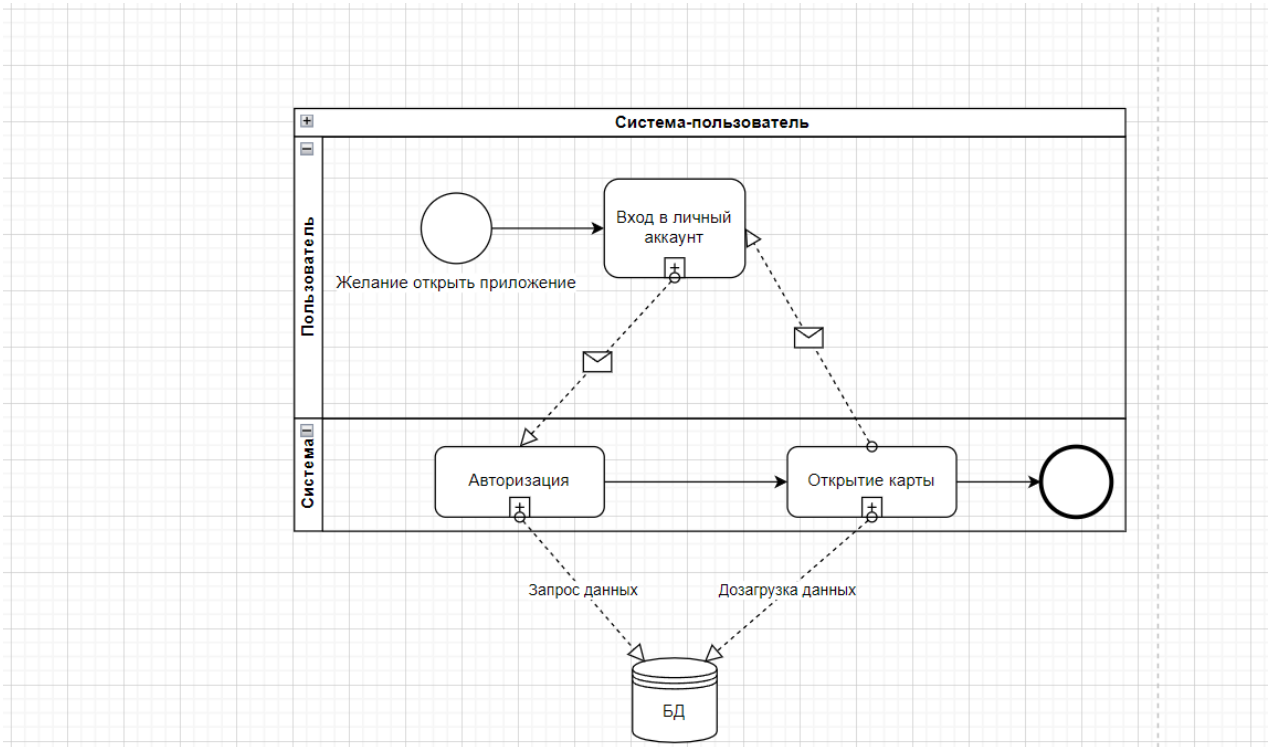


Рисунок 17 — Схема BPMN «Система-Пользователь»

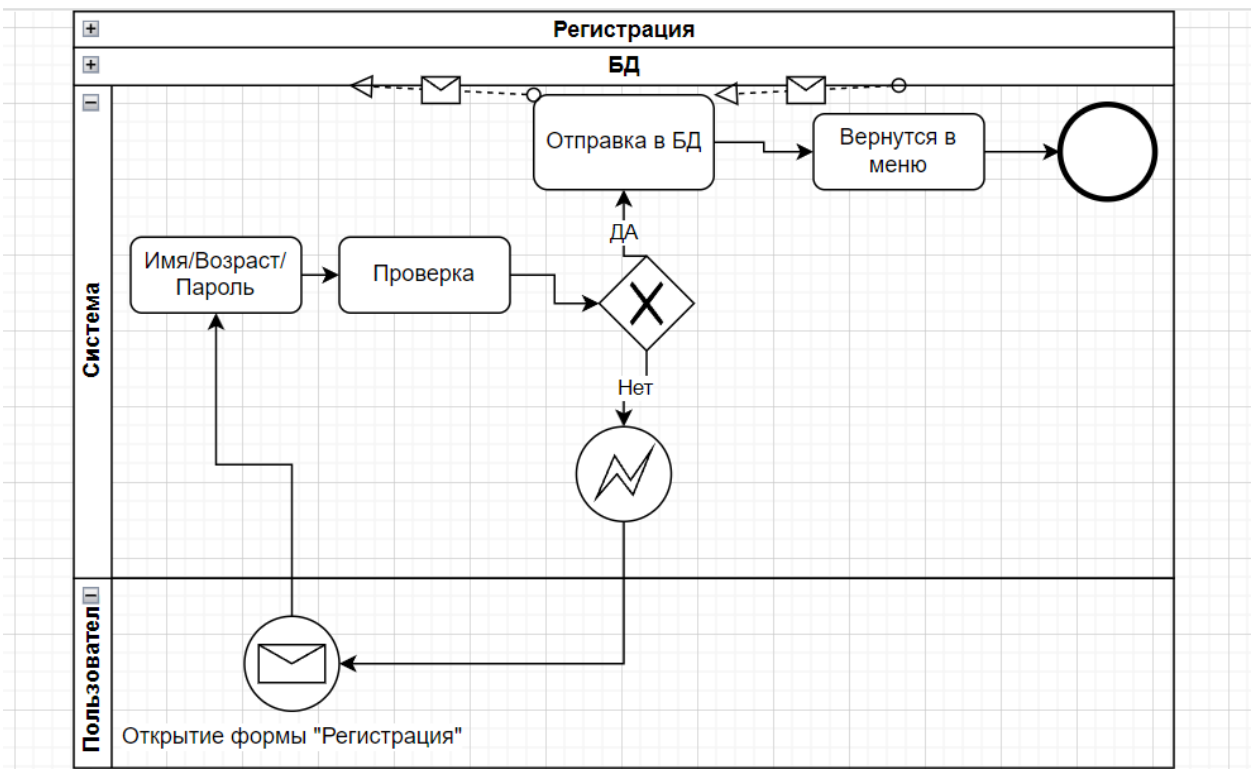


Рисунок 18 — Схема BPMN «Регистрация»

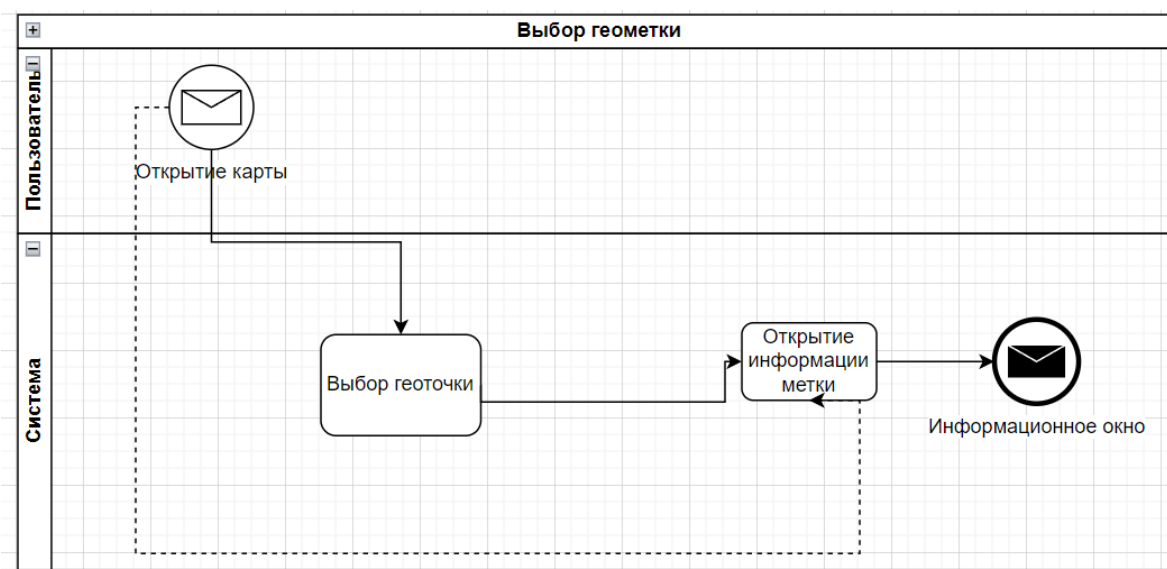


Рисунок 19 — Схема BPMN «Выбор геометки техники»

2.3 Объектно-ориентированное проектирование.

В методе объектно-ориентированного проектирования система рассматривается как набор объектов (сущностей). Состояние распределяется между объектами, и каждый объект обрабатывает данные своего состояния. Например, в ПО для автоматизации библиотек каждый представитель библиотеки может быть отдельным объектом со своими данными и функциями для работы с этими данными. Задачи, определенные для одной цели, не могут ссылаться или изменять данные других объектов. Объекты имеют свои внутренние данные, которые представляют их состояние. Подобные объекты создают класс. Другими словами, каждый объект является членом некоторого класса. Классы могут наследовать свойства суперкласса.

- **Объекты:** все объекты, участвующие в разработке решения, называются объектами. Например, в качестве объектов рассматриваются человек, банки, компания и пользователи. Каждая сущность имеет некоторые атрибуты, связанные с ней, и некоторые методы для выполнения над атрибутами.
- **Классы.** Класс — это обобщенное описание объекта. Объект является экземпляром класса. Класс определяет все атрибуты, которые может иметь объект, и методы, которые представляют функциональность объекта.
- **Сообщения:** объекты обмениваются сообщениями. Сообщения состоят из целостности целевого объекта, имени запрошенной операции и

любого другого действия, необходимого для выполнения функции. Сообщения часто реализуются как вызовы процедур или функций.

- Абстракция В объектно-ориентированном дизайне сложность решается с помощью абстракции. Абстракция — это удаление ненужного и усиление существенного.
- Инкапсуляция: Инкапсуляция также называется концепцией сокрытия информации. Данные и операции связаны с одним блоком. Инкапсуляция не только объединяет важную информацию об объекте, но и ограничивает доступ к данным и методам из внешнего мира.
- Наследование: OOD позволяет подобным классам выстраиваться в иерархическом порядке, при этом более низкие классы или подклассы могут импортировать, реализовывать и повторно использовать разрешенные переменные и функции из своих непосредственных суперклассов. Это свойство OOD называется наследованием. Это упрощает определение конкретного класса и создание обобщенных классов из конкретных.
- Полиморфизм: языки OOD предоставляют механизм, в котором методам, выполняющим аналогичные задачи, но различающимся аргументами, может быть присвоено одно и то же имя. Это известно как полиморфизм, который позволяет одному интерфейсу выполнять функции для разных типов. В зависимости от того, как вызывается служба, выполняется соответствующая часть кода.

Для данной работы были использованы модели на языке моделирования UML.

2.4 Диаграмма вариантов использования (USE-CASE).

Для моделирования системы наиболее важным аспектом является захват динамического поведения. Динамическое поведение означает поведение системы, когда она работает/ не работает.

Только статического поведения недостаточно для моделирования системы, скорее динамическое поведение важнее статического поведения. В UML доступно пять диаграмм для моделирования динамической природы, и диаграмма вариантов использования является одной из них.

Эти внутренние и внешние агенты известны как акторы. Диаграммы вариантов использования состоят из действующих лиц, вариантов использования и их взаимосвязей. Диаграмма используется для моделирования системы/подсистемы приложения.

Ключевой концепцией моделирования вариантов использования является то, что оно помогает нам проектировать систему с точки зрения конечного пользователя. Это эффективная техника для сообщения о поведении системы с точки зрения пользователя путем указания всего поведения системы, видимого извне. Сама диаграмма для разрабатываемой системы представлена на Рисунке 20.



Рисунок 20 — Диаграмма вариантов использования (USE-CASE)

2.5 Диаграммы последовательности.

Диаграмма последовательности представляет поток сообщений в системе и также называется диаграммой событий. Это помогает в представлении нескольких динамических сценариев. Он изображает связь между любыми двумя линиями жизни как упорядоченную во времени последовательность событий, так что эти линии жизни участвуют во время выполнения. В UML линия жизни представлена вертикальной чертой, тогда как поток сообщений представлен вертикальной пунктирной линией, проходящей через нижнюю часть страницы. Он включает в себя итерации, а также ветвление.

Сообщения, которыми обмениваются объекты, могут быть нескольких типов. Также сообщения могут быть настроены для отражения операций и свойств исходного и целевого элементов.

У диаграммы последовательности есть два измерения: вертикальное (время) и горизонтальное (объекты, участвующие во взаимодействии). Как правило, временная ось направлена сверху вниз (при желании направление осей можно поменять местами). Как правило, существенна именно последовательность сообщений, однако для приложений реального времени временная ось может служить реальной метрикой. Порядок горизонтального расположения объектов значения не имеет.

Диаграмма последовательности представлена на Рисунке 21.С другими диаграммами можно ознакомиться в Приложении В.

Используются следующие диаграммы:

Вход – используется в информационной системе для обработки запроса на предоставление доступа для выполнения определенных операций в самой системе. В данной диаграмме рассматривается авторизация пользователя приложения.

Последовательность функционирования диаграммы «Вход в приложение» для водителя системы:

1. Пользователь вводит данные в окно входа и нажимает кнопку вход
2. Иницируется проверка данных
3. Информация после проверки поступает в базу данных
4. Выполняется поиск пользователя в базе данных
5. Поток данных возвращается на сервер и последний подтверждает вход
6. Отображение формы с картой

Регистрация – используется в информационной системе для обработки запроса на создание учетной записи предоставление доступа для выполнения определенных операций в самой системе. В данной диаграмме, представленной на рисунке выше, рассматривается регистрация пользователя приложения.

Последовательность функционирования диаграммы «Регистрация» для пользователя системы:

1. Клиент вводит данные водителя в окно регистрации и сохраняет введенную информацию
2. Иницируется проверка данных
3. Информация поступает на главную форму
4. Выполняется сохранение в базе данных
5. Поток данных возвращается из базы данных и последняя подтверждает сохранение

6. Возврат на главную форму

Вывод информационного окна - Используется для отображения информации о технике для аудитории данного продукта.

На рисунке показана диаграмма последовательности «Вывод информационного окна». Данная диаграмма показывает последовательность действий для вывода окна с информацией, ради которой пользователь и заходил в приложение.

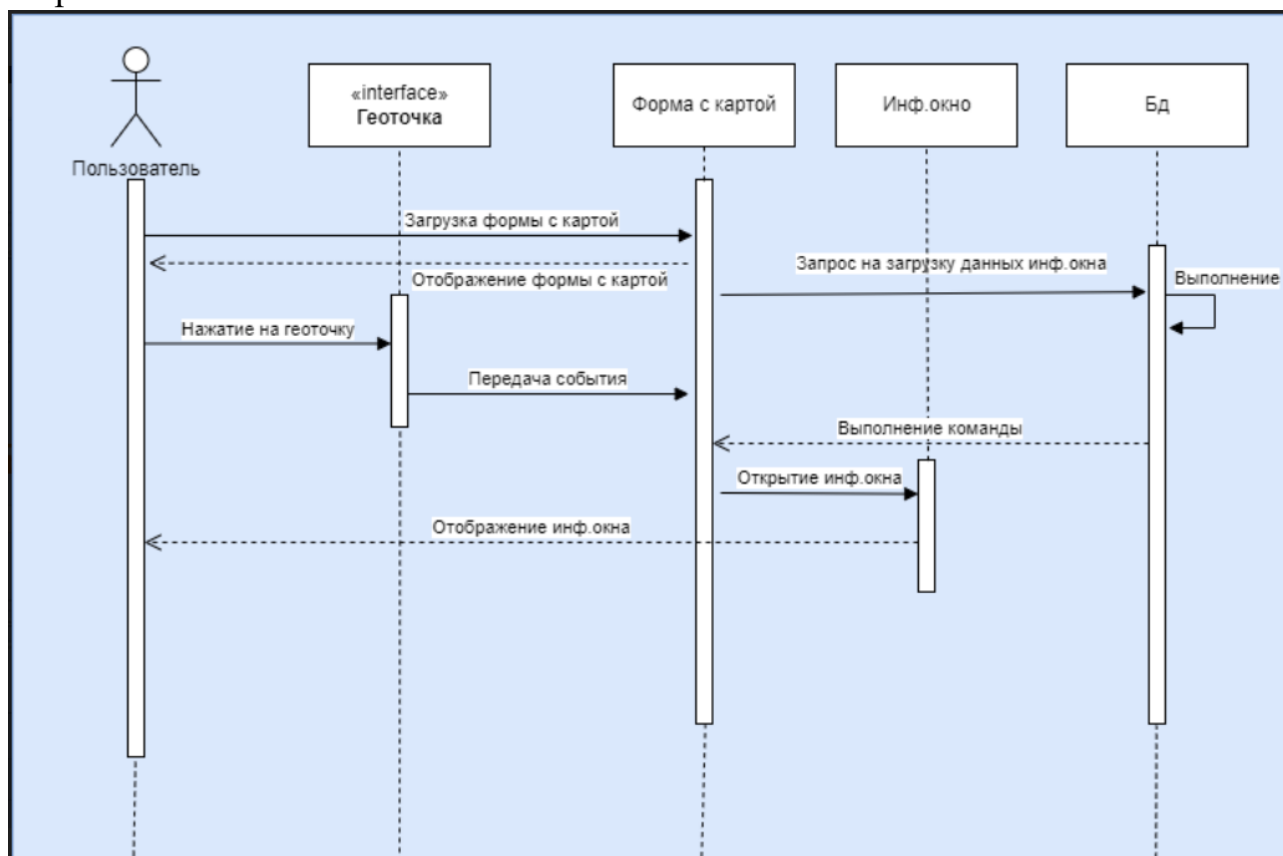


Рисунок 21 — Диаграмма последовательности для варианта использования «Вывод информационного окна»

Последовательность функционирования диаграммы «Вывод информационного окна» для пользователя системы:

1. Пользователь нажимает на геоточку с техникой
2. Передача команды форме с картой
3. Загрузка информации из базы данных
4. Открытие и отображение информационного окна

2.6 Диаграмма классов.

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Кроме того, диаграммы классов составляют основу еще

двух диаграмм – компонентов и развертывания. Диаграмма классов может отражать различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы.

Диаграмма классов может использоваться для отображения классов, отношений, интерфейса, ассоциации и совместной работы. UML стандартизирован в диаграммах классов. Поскольку классы являются строительным блоком приложения, основанного на ООП, диаграмма классов имеет соответствующую структуру для представления классов, наследования, отношений и всего того, что ООП имеют в своем контексте. Он описывает различные виды объектов и статические отношения между ними.

Каждый класс представлен прямоугольником, состоящим из трех частей: имени, атрибутов и операции.

Существует три типа модификаторов, которые используются для определения видимости атрибутов и операций.

- + используется для публичной видимости (для всех)
- # используется для защищенной видимости (для друзей и производных)
- используется для приватной видимости (только для меня)

Диаграмма классов данной информационной системы представлена на рисунке 22.

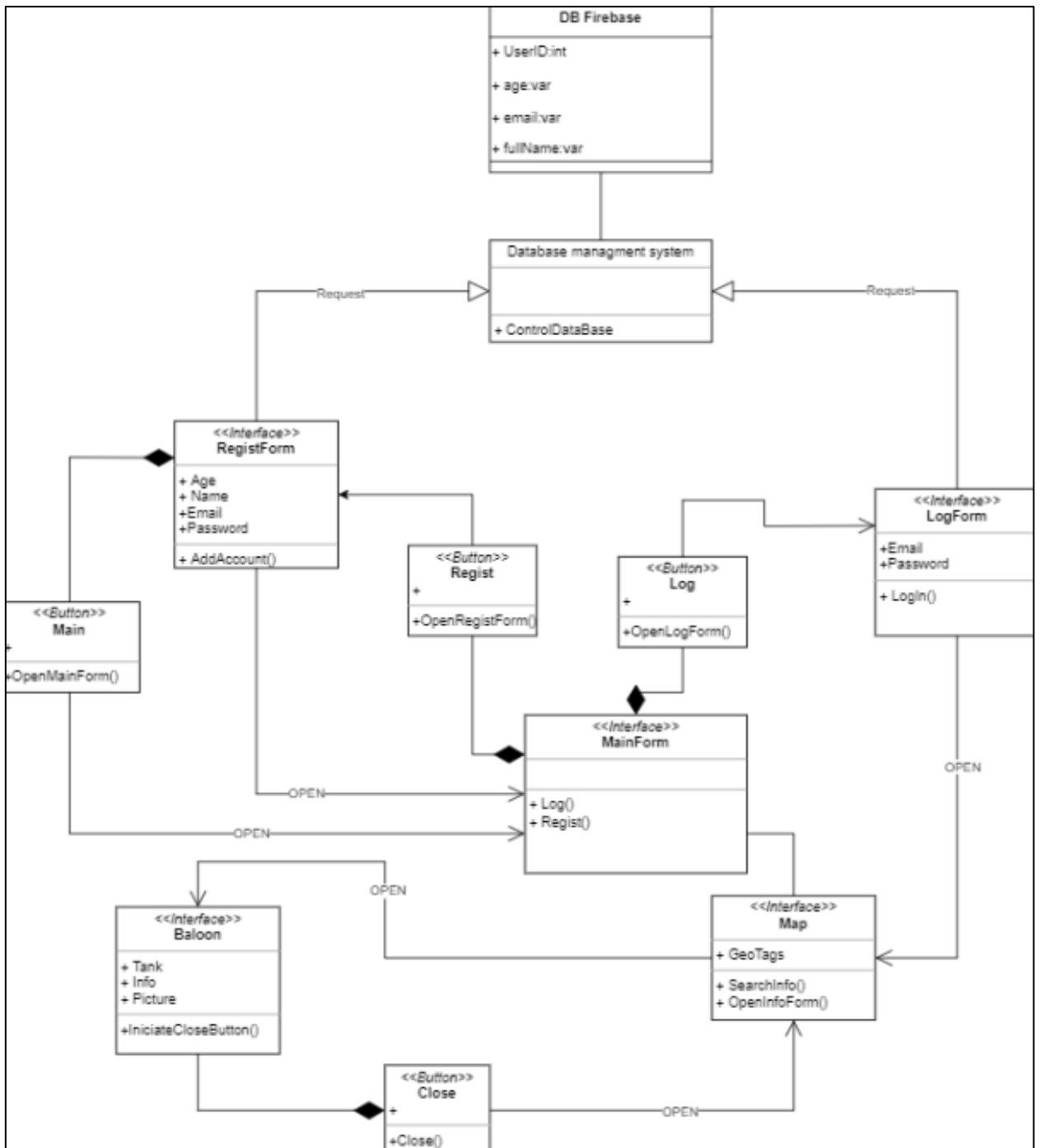


Рисунок 22 — Диаграмма классов

2.7 Диаграмма развёртывания.

Диаграммы развёртывания — это своего рода архитектурная модель, которая показывает статическую структуру запланированной целевой среды. Они показывают, как различные компоненты распределенной системы развёртываются на доступных аппаратных платформах (Рисунок.23)

При разработке диаграмм развёртывания преследуются следующие цели:

- специфицировать физические узлы, необходимые для размещения на них исполнимых компонентов программной системы;
- показать физические связи между узлами реализации системы на этапе ее исполнения;
- выявить узкие места системы и реконфигурировать ее топологию для достижения требуемой производительности.

Диаграммы развертывания иллюстрируют отношения между программными и аппаратными компонентами в системе и физическим распределением элементов системы. Обычно создаются во время реализации системы и определяют физическое расположение элементов процессора, программное обеспечение, установленное на каждом процессоре, а также компоненты и другие элементы, реализуемые программным обеспечением. Помимо процессоров узлы на диаграмме развертывания могут отображать аппаратные устройства, такие как датчики и принтеры, а также другие устройства, поддерживающие среду выполнения системы.

Обозначения, используемые на диаграммах развертывания, представляют узлы (обычно оборудование или виртуальные машины) в виде трехмерных блоков, которые могут содержать установленное программное обеспечение. Допускается использовать типы узлов для представления абстрактных развертываний или конкретных экземпляров узлов.

На практике диаграммы развертывания не используются повсеместно.

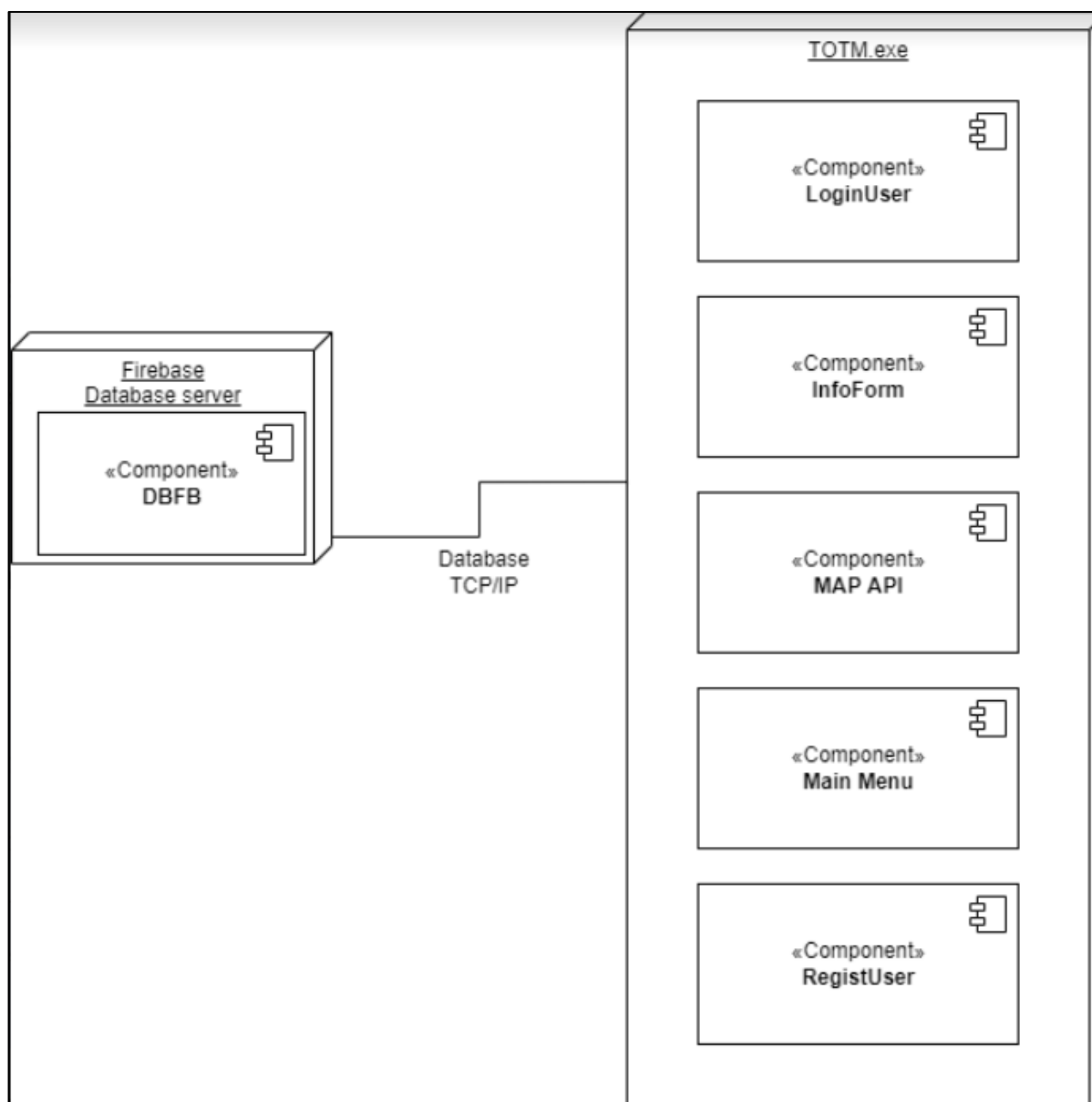


Рисунок 23 — Диаграмма развертывания

Процесс проектирования уже давно стал стандартом разработки ПО. Поэтому, чтобы не сталкиваться с превосходно написанной программой, которая из-за отсутствия нормальной документации обростаёт ненужным побочным функционалом, рудиментами, становится громоздкой и теряет свое качество, необходимо детально прорабатывать систему ещё до её реализации с помощью различных видов анализа, схем и диаграмм. После глубокого проектирования, составления схем и диаграмм дальнейшим шагом будет реализация разрабатываемой ИС.

ГЛАВА 3 РАЗРАБОТКА ИС

3.1 Основные элементы клиентского интерфейса.

После анализа, моделирования, проектирования и подготовки документации, следует инициализировать процесс разработки информационной системы. В данном случае процесс разработки начинается с создания интерфейса приложения.

Интерфейс, является ключевым компонентом связи пользователя с компьютером, основанный на представлении всех доступных пользователю системных объектов и функций в виде графических компонентов.

Программно интерфейс реализуется на базе библиотек Android Studio для языка программирования Java. Первичный вид для конвертации в код был также сделан в свободной среде для разработки приложений Android Studio.

Разработанный интерфейс программы (основные окна) показан на рисунках 24-27.

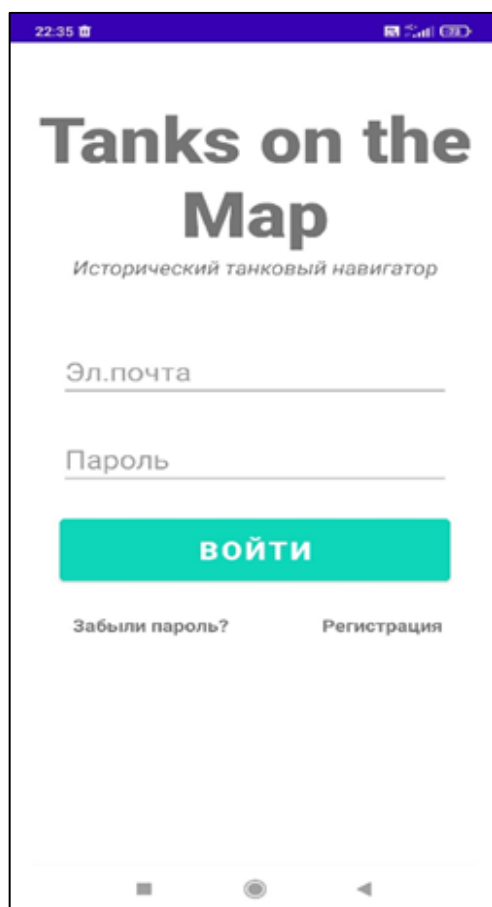


Рисунок 24 — Главное меню программы (Вход)

Приложение встречает нас окном входа, где пользователю требуется ввести свои данные в поля ввода. Если все указано верно, то пользователю становится доступна карта с геометками техники.

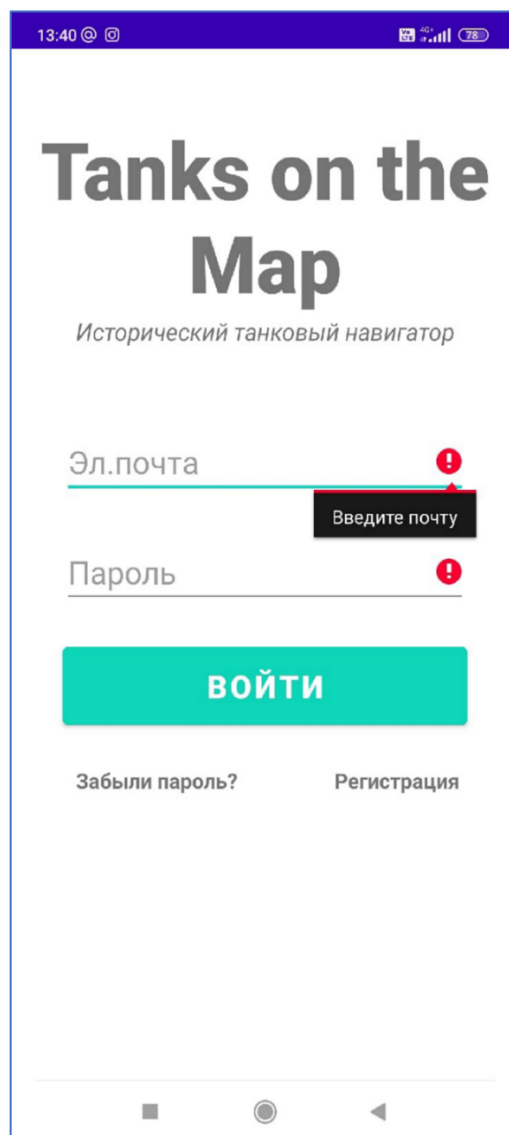


Рисунок 25 — Ошибка ввода данных(Вход)

В противном случае, если клиент впервые в данном приложении, придется нажать на кнопку «Регистрация». А в полях для вводимых данных высветятся сообщения об ошибках. Валидация является необходимым инструментом для опознания старых и новых пользователей. Проверка ввода пользователя является очень распространенным требованием в большинстве приложений.

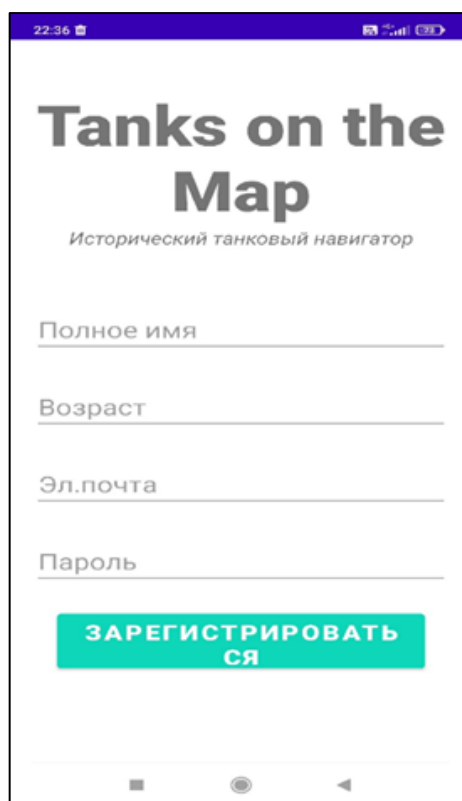
The image shows a mobile application registration screen. At the top, the title 'Tanks on the Map' is displayed in a large, bold, black font. Below the title, the subtitle 'Исторический танковый навигатор' is written in a smaller, italicized font. The form consists of four input fields: 'Полное имя', 'Возраст', 'Эл.почта', and 'Пароль'. Each field is a simple text box with a thin border. Below the 'Пароль' field is a prominent green button with white text that reads 'ЗАРЕГИСТРИРОВАТЬСЯ'. The entire form is centered on a white background. At the very top of the screen, a status bar shows the time '22:36' and various system icons.

Рисунок 26 — Форма регистрации

При переходе по кнопке в форму «Регистрация» пользователю предлагается заполнить небольшую анкету, при помощи которой он будет идентифицировать себя в данном приложении. Ему необходимо указать следующие данные:

- Полное имя (Имя пользователя для использования в приложении. Изменить его нельзя)
- Возраст (Необходим для определения основной группы(аудитории), которая использует данный продукт)
- Эл.почта
- Пароль (Указываются только цифры)

После указания всех корректных данных пользователя, после нажатия на кнопку «Зарегистрироваться», система уведомляет о успешной регистрации и тогда необходимо нажать на название приложения для возврата в меню входа.

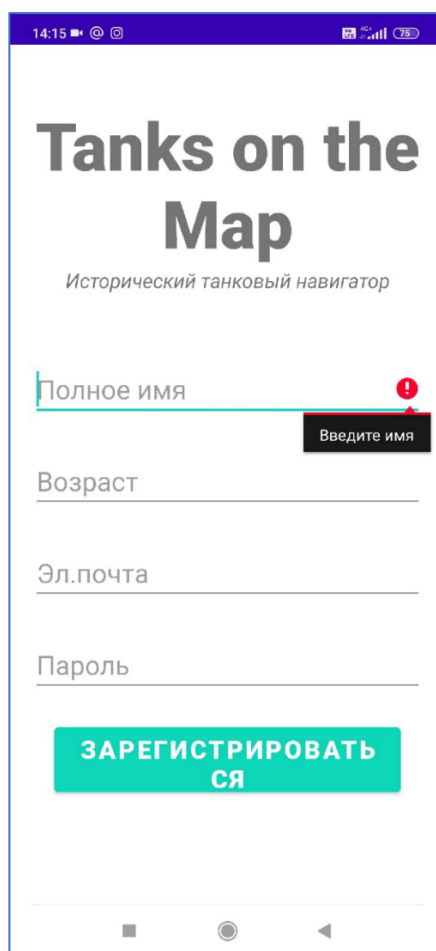


Рисунок 27 — Ошибка ввода данных(Вход)

Как и с главным меню, если данные введены неверно, то пользователя уведомят об ошибке введенной им информации.

При попадании на карту взору пользователя предстанет интерактивная карта с предустановленными на ней метками бронетехники. Метки расположены в странах производителях конкретной боевой машины. При нажатии на метку происходит отображение названия техники с последующей актуальной информацией.

По желанию пользователя, можно свернуть приложение и обратиться к классическому сервису Google Maps. Для использования данной функции предустановлены необходимые кнопки в правом нижнем углу экрана.

3.2 Технологические процессы, обрабатывающие данные.

Технологический процесс обработки данных — это комплекс взаимосвязанных операций направленных на преобразование информации с этапа ее появления до этапа ее перехода к конечному пользователю.

В качестве базы данных на данный момент выбрана Firebase Realtime Database – облачная база данных корпорации Google.

Основной сервис — облачная СУБД класса NoSQL, позволяющая разработчикам приложений хранить и синхронизировать данные между несколькими пользователями. Поддержаны особенности интеграции с приложениями под операционные системы Android и iOS, реализовано API для приложений на Java также возможно работать напрямую с базой данных в стиле REST из ряда JavaScript-фреймворков, включая AngularJS, React, Vue.js, Ember.js и Backbone.js. Предусмотрено API для шифрования данных. База представлена на рисунках 28 и 29.

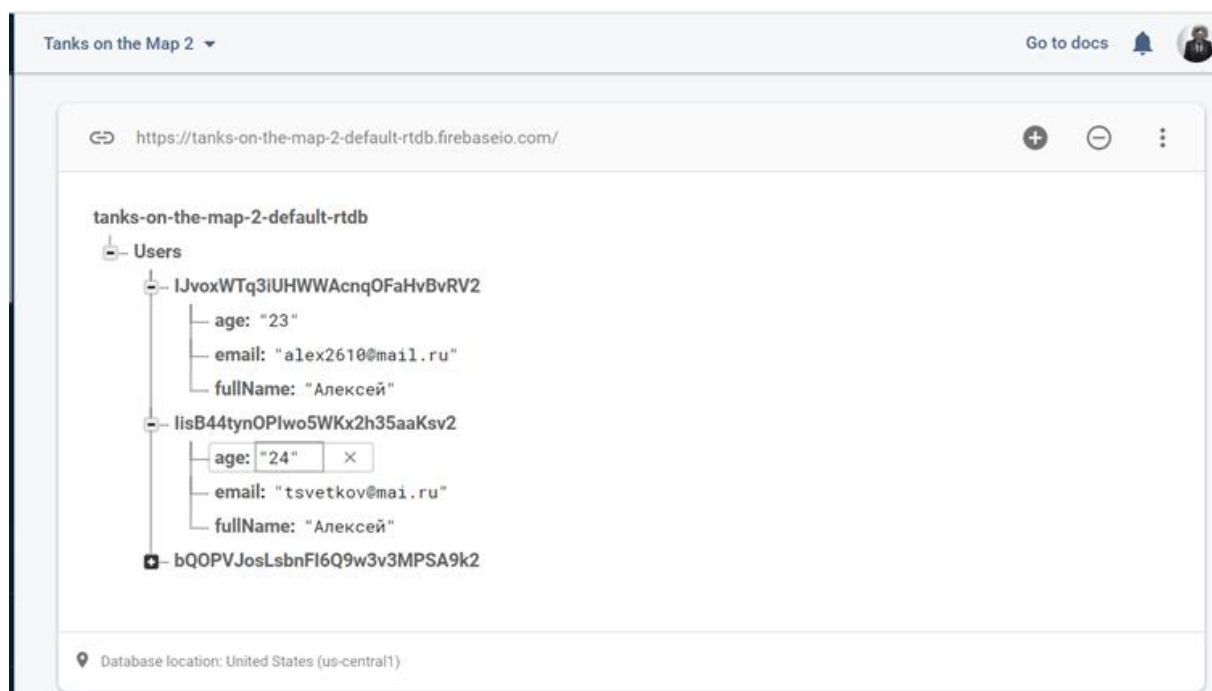


Рисунок 28 — Содержимое данных

Identifier	Providers	Created ↓	Signed In	User UID
arabico@mail.ru	✉	Dec 23, 2021	Dec 24, 2021	bQOPVJosLsbnFi6Q9w3v3MPSA9...
tsvetkov@mai.ru	✉	Dec 21, 2021	Dec 21, 2021	iisB44tynOPlwo5WKx2h35aaKsv2
alex2610@mail.ru	✉	Dec 21, 2021	Dec 21, 2021	IJvoxWTq3iUHWWAcnqOFaHvBvR...

Рисунок 29 — Данные пользователей

3.3 Схема восстановления и сохранения информации.

В процессе эксплуатации информационных систем может происходить разрушение информации по причинам:

- физическое разрушение носителя информации или отдельных частей носителя с информацией;
- аппаратные сбои и разрушение информации;
- программное разрушение информации из-за ошибок в программах;
- воздействие вредоносных программ;
- ошибки оператора (персонала);
- +/- преднамеренное разрушение с использованием различных средств.

Поэтому, обязательным элементом любых ИС являются процедуры сохранения и восстановления информации как защитная мера от разрушения (потери) данных.

План аварийного восстановления (DRP) — это документированный структурированный подход, описывающий, как организация может быстро возобновить работу после незапланированного инцидента. DRP является неотъемлемой частью плана обеспечения непрерывности бизнеса. Он применяется к аспектам организации, которые зависят от функционирующей ИТ-инфраструктуры. Задача DRP — помочь организациям (и разработчикам в частности) справиться с потерей данных и восстановить функциональность системы, чтобы она могла работать после инцидента, даже если она работает на минимальном уровне.

Пошаговый план состоит из мер предосторожности, направленных на минимизацию последствий стихийного бедствия, чтобы организация могла продолжить работу или быстро возобновить выполнение критически важных функций. Как правило, планирование аварийного восстановления включает анализ бизнес-процессов и потребностей в непрерывности. Перед созданием подробного плана организация часто проводит анализ влияния на бизнес и анализ рисков, а также устанавливает цели восстановления.

Поскольку киберпреступность и нарушения безопасности становятся все более изощренными, для организации важно определить свои стратегии восстановления и защиты данных. Способность быстро обрабатывать инциденты может сократить время простоя и свести к минимуму как финансовый, так и репутационный ущерб. Кроме того, DRP позволяют организациям убедиться, что они соответствуют всем требованиям соответствия, а также предоставляют четкую дорожную карту для восстановления.

Стратегия аварийного восстановления должна начинаться на бизнес-уровне и определять, какие приложения наиболее важны для работы

организации. Целевое время восстановления описывает целевое время, в течение которого бизнес-приложение может быть недоступно, обычно измеряемое в часах, минутах или секундах. Целевая точка восстановления описывает возраст файлов, которые необходимо восстановить из хранилища резервных копий, чтобы возобновить нормальную работу.

Стратегии восстановления определяют планы организации по реагированию на инцидент, а планы аварийного восстановления описывают, как организация должна реагировать.

DRP могут быть специально адаптированы для данной среды. Некоторые планы, ориентированные на окружающую среду, включают:

Виртуализированный план аварийного восстановления.

Виртуализация предоставляет возможности для реализации аварийного восстановления более эффективным и простым способом. Виртуализированная среда может запускать новые экземпляры виртуальных машин за считанные минуты и обеспечивать восстановление приложений благодаря высокой доступности.

План аварийного восстановления сети.

Разработка плана восстановления сети усложняется по мере увеличения сложности сети. Важно подробно описать пошаговую процедуру восстановления, правильно протестировать ее и постоянно обновлять. Данные в этом плане будут специфичны для сети, например, по ее производительности и сетевому персоналу.

План аварийного восстановления в облаке.

Аварийное восстановление в облаке (облачное аварийное восстановление) может варьироваться от резервного копирования файлов в облаке до полной репликации. Облачное аварийное восстановление может быть эффективным с точки зрения пространства, времени и затрат, но для поддержания плана аварийного восстановления требуется надлежащее управление.

План аварийного восстановления центра обработки данных.

Этот тип плана ориентирован исключительно на объекты и инфраструктуру центра обработки данных. Оценка операционного риска является ключевым элементом центра обработки данных. Он анализирует ключевые компоненты, такие как местоположение здания, системы электропитания и защиты, безопасность и офисные помещения.

Процесс планирования аварийного восстановления включает в себя больше, чем просто написание документа.

Контрольный список DRP должен включать следующие шаги:

- установление диапазона или степени необходимого лечения и деятельности -- объем выздоровления;
- сбор соответствующих документов по сетевой инфраструктуре;
- выявление наиболее серьезных угроз и уязвимостей
- определение текущих стратегий аварийного восстановления;
- определение группы реагирования на инциденты;
- рассмотрение и утверждение руководством DRP;
- тестирование плана;
- обновление плана; и
- проведение аудита DRP.

Пользователи Firebase могут настроить свою базу данных Firebase Realtime для автоматического резервного копирования, функцию самообслуживания, которая обеспечивает ежедневное резервное копирование данных и правил приложения базы данных в формате JSON в корзину облачного хранилища (Рисунок 30).

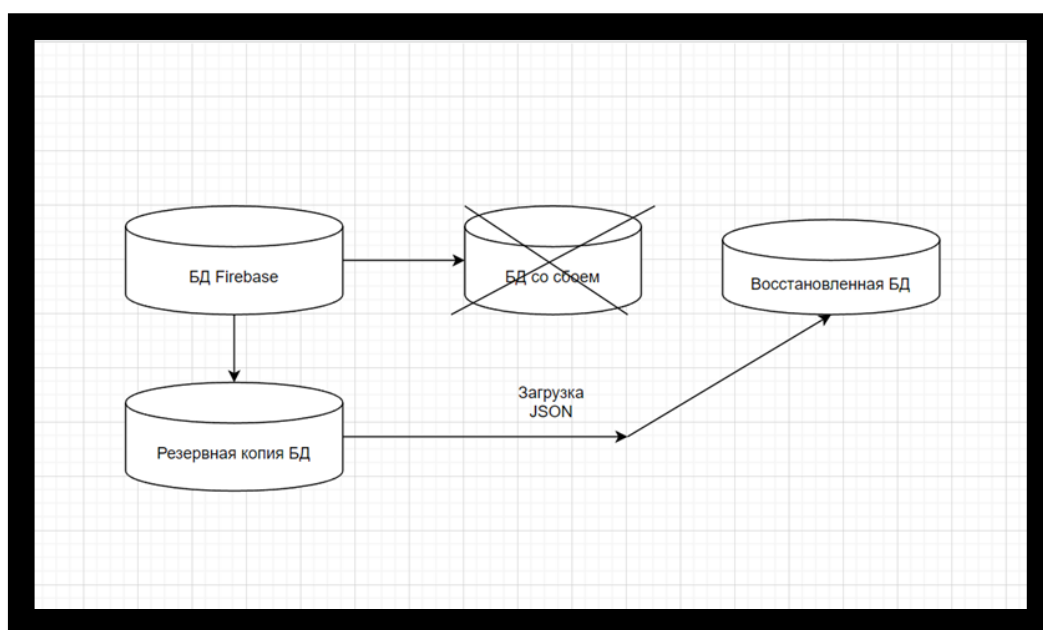


Рисунок 30 — Схема восстановления базы данных

3.4 Экономическая эффективность

Существует ряд факторов, влияющих на стоимость разработки. Она зависит от сложности проекта, ведь на основании этого подбирается команда исполнителей и рассчитывается количество часов, которые специалисты потратят на разработку. Большое значение имеет содержание услуг. Для успешного запуска и дальнейшего функционирования программы, помимо написания программного кода, необходимы аналитический анализ, создание оригинального дизайна, тестирование и прочее. В-третьих, существует такая статья расходов, как развитие проекта после релиза.

В целом, количество денег и реального времени рассчитываются на основании нижеописанных аспектов. (Рисунок 31)



Рисунок 31 — Пример схемы затрат на разработку

При расчете сметы обязательно нужно учитывать количество часов, необходимое на разработку. Соответственно, чем труднее задача, тем больше времени потребуется на ее выполнение, и тем выше будет цена.

Компоненты, которые требуют наибольших трудозатрат:

- Проектирование
- Дизайн интерфейса

Этому этапу стоит уделить внимание, так как от интерфейса мобильного приложения будет напрямую зависеть удобство его использования и ощущения пользователей. Неудобно расположенные элементы, неявные и непонятные функции могут существенно снизить удобство использования мобильного приложения.

Дизайн тоже играет большую роль, так как встречают по одежке. Невзрачное приложение с неудобным интерфейсом почти гарантированно оттолкнет будущих пользователей и заставит посмотреть в сторону решений конкурентов.

-Разработка функционала приложения (особенно клиент-серверных)

Самый важный этап разработки, так как по сути все полезные функции приложения находится тут. То какой функционал будет заложен в приложении и каким образом он будет реализован, напрямую влияет на трудозатраты и цену приложения.

-Бэкенд

Для работы большинства приложений необходимо наличие серверной системы для хранения базы данных, реализации бизнес-логики проекта и передачи запрашиваемой информации клиентской стороне в соответствии с API. Созданием этого компонента занимаются бэкенд-специалисты.

-Административная панель

Без такого элемента невозможно управление контентом и пользовательскими аккаунтами. Он требует многоэтапной разработки с формированием API и бэкенда, который будет отвечать за прием информации от приложения, хранение и обработку информации, а также передачу информации в приложение.

-Разные виды пользователей

В зависимости от типа приложения может понадобиться разработка нескольких интерфейсов. Самым простым примером являются программы вроде Uber, то есть к примеру, которыми одновременно пользуется и заказчик, и исполнитель. Соответственно, для них нужно создать отдельный функционал, продумать визуальную составляющую, бизнес-логику, сверстать экраны и так далее.

В целом, количество денег и реального времени рассчитываются на основании нижеописанных аспектов.

Трудоемкость разработки программного изделия (t) определяется по формуле, чел.-мес.:

$$t = 3,6(n_{\text{т.и.к.}})^{1,2}, (1)$$

где $n_{\text{т.и.к.}}$ — число тысяч исходных команд.

Продолжительность разработки программного изделия (T) рассчитывается по формуле, мес.:

$$T = 2,5 * t^{0,32}, (2)$$

Среднее число исполнителей (чи) рассчитывается исходя из определенных или заданных характеристик трудоемкости и длительности разработки программного изделия по формуле, чел.:

$$\text{чи} = \frac{t}{T}, (3)$$

Все затраты труда суммируются, т.е.

$$t = t_o + t_{и} + t_a + t_{п} + t_{отл} + t_d, (4)$$

Условное число операторов Q в программе задачи определяется по формуле:

$$Q = qc(1 + p), \text{ где } (5)$$

q — предполагаемое число операторов;

c — коэффициент сложности программы, принимает значение 1,25;

p — коэффициент коррекции программы, принимает значение 0,05.

Затраты труда на изучение описания задачи и рассчитываются по формуле, чел.-ч.:

$$t_{и} = \frac{Q \cdot B}{\left(\frac{75}{85}\right)^k} \text{ где } (6)$$

B — коэффициент, учитывающий качество постановки задачи выданной для разработки программы, принимается равным 1,2

k — коэффициент квалификации разработчика, принимает разные значения в зависимости от стажа работы (до 2-х лет — 0,3; 2-3 года — 1,0; 3-5 лет — 1,1..1,2; 5-7 лет — 1,3..1,4; свыше 7 лет — 1,5..1,6).

$$t_a = \frac{Q}{\left(\frac{20}{25}\right)^k} (7)$$

Затраты труда на составление программы $t_{п}$ рассчитываются по формуле, чел.-ч.:

$$t_{п} = \frac{Q}{\left(\frac{20}{25}\right)^k} (8)$$

Затраты труда на отладку программы на ЭВМ $t_{отл}$ рассчитываются по следующим формулам, чел.-ч.:

$$t_{отл} = \frac{Q}{\left(\frac{4}{5}\right)^k} (9)$$

Затраты труда на подготовку документации по задаче t_d определяются по формуле, чел.-ч.:

$$t_d = t_{др} + t_{до}, \text{ где } (10)$$

$t_{др}$ — затраты труда на подготовку материалов в рукописи, равная $Q / (15 \div 20) k$;

$t_{до}$ — затраты труда на редактирование, печать, оформление документации, равные $0,75 t_{др}$. Результаты представлены на рисунке 32.

$n_{т.и.к.}$	10	Число тысяч исходных команд			
t	57,05615	чел.-мес.		$k=$	0,3
T	9,119201	Мес.		Зарплата в месяц:	45000
P_p	175,2659	исх. Команд/чел.-мес.		Часов работы (в мес.):	120
$Ч_{и=}$	1	чел.		Оплата за час труда:	375
t_0	320	чел.-ч.		Оплата программисту:	20 507,81 Р
$t_{и=}$	59,5	чел.-ч.		Дней:	6,8359375
t_3	54,6875	чел.-ч.		Недель:	0,9765625
t_n	54,6875	чел.-ч.			
$t_{отл=}$	54,6875	чел.-ч.		Суммарные затраты:	242 117,19 Р
t_d	102,0833	чел.-ч.			
t	645,6458	чел.-ч.			
q	10	Предполагаемое число операторов			
c	1,25	Коэффициент сложности программы			
p	0,05	Коэффициент коррекции программы			
Q	13,125	Условное число операторов			
B	1,2	Коэффициент качества			

Рисунок 32 — Расчет трудоемкости (по формулам)

3.5 Сроки окупаемости.

Предположим, что используются бесплатные версии дополнительных обслуживающих программ (база данных, средство разработки и др.). Тогда затратами будет являться только оплата труда разработчика (мы предполагаем, что приложение не будет получать обновление, а её поддержка будет бесплатна, т.е. без найма ответственного за это лица).

Далее определяется, из чего получается выгода, используя такую систему. Программа направлена на уменьшение времени поиска информации о бронетанковой технике. Следовательно, если пользователи предлагают за использование фиксированную сумму пожертвований и определённый срок, за который мы укладываемся, реализуя новые возможности, то можно сэкономить на рабочем труде или начать уже новый проект.

Так, можно найти окупаемость, исходя из среднего размерами пожертвоаний (как на Patreon)(если учитывать такой вид оценки, то сроки окупаемости могут быть от 6 до 12 месяцев, в зависимости от клиентской базы приложения) или исходя из общих сроков окупаемости.

На практике же доход поступает совсем не равными частями. Более того, ценность денег во времени меняется Фактор времени в простом сроке окупаемости не учитывается. Для расчета срока окупаемости с учетом

изменения стоимости денег во времени и используется дисконтированный способ расчета.

$$DPP = \sum_{i=1}^n CF_i / (1+r)^i > IC, \text{ где (11)}$$

DPP - дисконтированный срок окупаемости, выраженный в годах/ месяцах;

CF – ожидаемые денежные поступления в конкретный период;

IC – сумма первоначальных инвестиций;

r – процентная ставка;

n – количество лет (месяцев) окупаемости.

Результат расчета по формулам представлен на рисунке 33.

Год	CF	PV	CFxPV	CCF
0	1000000	1	1000000	1000000
1	100000	0.91	90909.09	909090.91
2	110000	0.83	90909.09	818181.82
3	120000	0.75	90157.78	728024.04
4	90000	0.68	61471.21	666552.83
5	100000	0.62	62092.13	604460.7

Дисконтированный период окупаемости = 14.73 лет

Рисунок 33— Сроки окупаемости

DPP позволяет учесть динамику стоимости денег, а также использовать для разных периодов разные нормы дисконта. В то же время, он имеет свои недостатки. К ним можно отнести невозможность учета потоков средств после достижения точки безубыточности, а также некорректные результаты при расчетах потоков с различными знаками (отрицательных и положительных).

Определение показателя DPP при оценке инвестиционного предложения позволяет снизить риски потерь инвестированных денег и в общем оценить ликвидность инициативы. Вместе с тем, по одному этому критерию делать далеко идущие выводы не следует, лучше всего оценить предложенный проект по ряду показателей и сделать вывод по всему комплексу полученных данных.

ЗАКЛЮЧЕНИЕ

Разработка ИС охватывает все процессы создания системы, начиная с предпроектного анализа, заканчивая сопровождением уже готового приложения.

В процессе выполнения ВКР были решены следующие задачи:

- Проведен анализ информационной системы и выбранной области с выделением требований;
- Проведен анализ типовых проектных решений с целью использования в собственных проектах, а также подобрать необходимые инструменты для дальнейшей разработки;
- Спроектирована информационная система с учетом проведенного комплексного анализа, с дальнейшей разработкой прототипа;

В ходе выполнения данной работы были сделаны следующие выводы:

- Среда разработки Android Studio является оптимальным решением для проектирования Android – приложения ввиду простоты работы и гибкого набора инструментов
- Облачная база данных Firebase Realtime Database корпорации Google отвечает всем требованиям для хранения, изменения сбора информации для выполнения установленных задач.
- Разработанная ИС ускоряет поиск информации при помощи карты, визуализируя информацию.
- Информация в проектируемой ИС соответствует историческим документам и повышает историческую грамотность

В результате работы были выполнены поставленные задачи, благодаря которым была разработана геоинформационная система «Приложение по поиску танков на карте», с дружелюбным интерфейсом и актуальной информацией, локацией бронемашин и с данными, с помощью которых можно найти интересующую пользователя технику.

Список использованных источников и литературы

1. Интернет-музей танков [Электронный источник] <https://tanks-encyclopedia.com/> [Дата обращения 01.04.2021]
2. Вендров А. М. Практикум по проектированию программного обеспечения экономических информационных систем. М.: Финансы и статистика, 2006.
3. Имперский военный музей [Электронный источник] <https://www.iwm.org.uk/> [Дата обращения 02.04.2021]
4. Ипатова Э.Р. Методологии и технологии системного проектирования информационных систем: Учебник / Э.Р. Ипатова, Ю.В. Ипатов; РАО. - М.: Флинта: МПСИ, 2008. - 256 с.:
5. Quora (автор статьи : Thomas Zinser) [Электронный источник] <https://www.quora.com/Why-do-you-like-tanks> [Дата обращения 03.04.2021]
6. Огоркевич Ричард Танки: 100 лет истории, 2019. - 336 с
7. 7. НОУ ИНТУИТ Виды диаграмм UML [Электронный источник] // Intuit URL: <https://intuit.ru/studies/courses/1007/229/lecture/5954?page=2> [дата обращения 06.04.2021]
8. УГОЛОК НЕБА Большая авиационная библиотека [Электронный источник] <http://www.airwar.ru/> [Дата обращения 11.04.2021]
9. M1A2 Abrams Main Battle Tank [Электронная статья] <https://www.military.com/equipment/m1a2-abrams-main-battle-tank> [Дата обращения 12.11.2021]
10. UML — диаграмма вариантов использования (use case diagram) [Электронный ресурс] / кукара4а — Электрон. ст. — Хабр — URL: <https://habr.com/ru/post/47940/>, свободный. — Яз. рус. — (Дата обращ. 03.05.2021).
11. [Электронный источник] // GeeksforGeeks URL: <https://www.geeksforgeeks.org/what-is-information-retrieval/> [дата обращения 29.09.2021]
12. Электронный фонд правовых и нормативно-технический документов [Электронный источник] // docs.cntd URL: <https://docs.cntd.ru/document/1200144954> [дата обращения 30.11.2021]

13. Элементы графической нотации диаграммы последовательности [Электронный источник] // Intuit URL: <https://intuit.ru/studies/courses/32/32/lecture/1014> [дата обращения 08.12.2021]
14. Заботина Н.Н. Проектирование информационных систем: Учебное пособие / Н.Н. Заботина. - М.: ИНФРА-М, 2011. - 331 с.
15. Применение UML и шаблонов проектирования. 2-е издание. / Ларман, Крэг.- Пер. с англ. - СПб: Символ-Плюс, 2004. - 192 с.,ил
16. disaster recovery plan (DRP) [Электронный источник] <https://www.techtarget.com/searchdisasterrecovery/definition/disaster-recovery-plan> [Дата обращения 12.12.2021]
17. Калькулятор расчета дисконтированного срока окупаемости [Электронный источник] <https://wpcalc.com/discounted-payback-period/> [Дата обращения 12.12.2021]
18. What is information availability and information unavailability? Explain BC planning life cycle [Электронный источник] <https://www.ques10.com/p/20620/what-is-information-availability-and-information-u/> [дата обращения 15.12.2021]
19. Что такое диаграмма Ганта? [Электронный источник] <https://www.atlassian.com/ru/agile/project-management/gantt-chart> [дата обращения 15.12.2021]
20. Интернет энциклопедия Armed Forces Survival [Электронный источник] <https://www.armedforcesmuseum.com/> [дата обращения 16.12.2021]

ПРИЛОЖЕНИЕ А

Код главной формы

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   tools:context=".MainActivity">
7
8   <TextView
9     android:id="@+id/banner"
10    android:layout_width="match_parent"
11    android:layout_height="wrap_content"
12    android:layout_marginTop="50dp"
13    android:fontFamily="sans-serif-black"
14    android:text="@string/banner"
15    android:textAlignment="center"
16    android:textSize="60sp"
17    android:textStyle="bold" />
18
19   <TextView
20     android:id="@+id/bannerDiscription"
21     android:textStyle="italic"
22     android:textSize="18sp"
23     android:textAlignment="center"
24     android:layout_below="@id/banner"
25
26     android:layout_width="match_parent"
27     android:layout_height="wrap_content"
28     android:text="@string/bannerDiscription" />
29
```

Рисунок А.1 — Код Main Activity

```
30 <EditText
31     android:id="@+id/email"
32     android:layout_width="380dp"
33     android:layout_height="wrap_content"
34     android:layout_below="@id/bannerDiscription"
35     android:layout_marginHorizontal="40sp"
36     android:layout_marginTop="60sp"
37     android:ems="10"
38     android:hint="@string/hint_email"
39     android:inputType="textEmailAddress"
40     android:textSize="24sp"
41     tools:ignore="TextContrastCheck,TextContrastCheck"
60     android:layout_width="match_parent"
61     android:layout_height="70dp"
62     android:layout_below="@id/password"
63     android:layout_marginHorizontal="40sp"
64     android:layout_marginTop="24sp"
65     android:backgroundTint="#0ED6B9"
66     android:text="@string/signIn"
67     android:textSize="24sp"
68     android:textStyle="bold"
69     tools:ignore="TextContrastCheck" />
52     android:hint="@string/password"
53     android:importantForAutofill="no"
54     android:inputType="textPassword"
55     android:textSize="24sp"
56     tools:ignore="TextContrastCheck,TextContrastCheck" />
57
58 <Button
```

Рисунок А.2 — Код Main Activity

```
56 tools:ignore="TextContrastCheck,TextContrastCheck"
57
58 <Button
59     android:id="@+id/signIn"
60     android:layout_width="match_parent"
61     android:layout_height="70dp"
62     android:layout_below="@id/password"
63     android:layout_marginHorizontal="40sp"
64     android:layout_marginTop="24sp"
65     android:backgroundTint="#0ED6B9"
66     android:text="@string/signIn"
67     android:textSize="24sp"
68     android:textStyle="bold"
69     tools:ignore="TextContrastCheck" />
70
71 <TextView
72     android:id="@+id/forgotPassword"
73     android:layout_width="wrap_content"
74     android:layout_height="wrap_content"
75     android:layout_below="@id/signIn"
76     android:layout_marginHorizontal="50dp"
77     android:layout_marginTop="25dp"
78     android:text="@string/forgotPassword"
79     android:textSize="16sp"
80     android:textStyle="bold"
81
82     />
83
84 <TextView
85     android:id="@+id/register"
```

Рисунок А.3 — Код Main Activity

ПРИЛОЖЕНИЕ Б

Код программы

```

1 package com.example.tanksonthemap;
2
3 import ...
22
23 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
24
25     private TextView register;
26     private EditText editTextEmail, editTextPassword;
27     private Button signIn;
28
29     private FirebaseAuth mAuth;
30     private ProgressBar progressBar;
31
32     @Override
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.activity_main);
36
37         register=(TextView)findViewById(R.id.register);
38         register.setOnClickListener(this);
39
40         signIn = (Button) findViewById(R.id.signIn);
41         signIn.setOnClickListener(this);
42         editTextEmail = (EditText) findViewById(R.id.email);
43         editTextPassword = (EditText) findViewById(R.id.password);
44         progressBar = (ProgressBar) findViewById(R.id.progressBar);
45         mAuth = FirebaseAuth.getInstance();
46
47     }
48
49     @Override

```

Рисунок Б.1 — Код Main Activity.Java

```

49     @Override
50     public void onClick(View v) {
51         switch (v.getId()) {
52             case R.id.register:
53                 startActivity(new Intent( packageContext this, RegisterUser.class));
54                 break;
55
56             case R.id.signIn:
57                 userLogin();
58                 break;
59         }
60     }
61
62     private void userLogin() {
63         String email = editTextEmail.getText().toString().trim();
64         String password = editTextPassword.getText().toString().trim();
65
66         if(email.isEmpty()){
67             editTextEmail.setError("Введите почту");
68             editTextEmail.requestFocus();
69             return;
70         }
71         if(!Patterns.EMAIL_ADDRESS.matcher(email).matches()){
72             editTextEmail.setError("Пожалуйста, укажите корректную почту");
73             editTextEmail.requestFocus();
74             return;
75         }
76         if(password.isEmpty()){
77             editTextPassword.setError("Введите пароль");
78             editTextPassword.requestFocus();
79             return;

```

Рисунок Б.2 — Код Main Activity.Java

```
74         return;
75     }
76     if(password.isEmpty()){
77         editTextPassword.setError("Введите пароль");
78         editTextPassword.requestFocus();
79         return;
80     }
81     if(password.length() < 6){
82         editTextPassword.setError("Пароль должен состоять из 6 символов");
83         editTextPassword.requestFocus();
84         return;
85     }
86
87     progressBar.setVisibility(View.VISIBLE);
88     mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
89         @Override
90         public void onComplete(@NonNull Task<AuthResult> task) {
91
92             if(task.isSuccessful()){
93                 startActivity(new Intent( packageContext, MainActivity.this, MapsActivity.class));
94             }else{
95                 Toast.makeText( context, MainActivity.this, text: "Ошибка входа!Пожалуйста проверьте введенные данные", Toast
96             }
97         }
98     });
99
100 }
101 }
102
```

Рисунок Б.3 — Код Main Activity.Java

ПРИЛОЖЕНИЕ В

Диаграммы последовательности

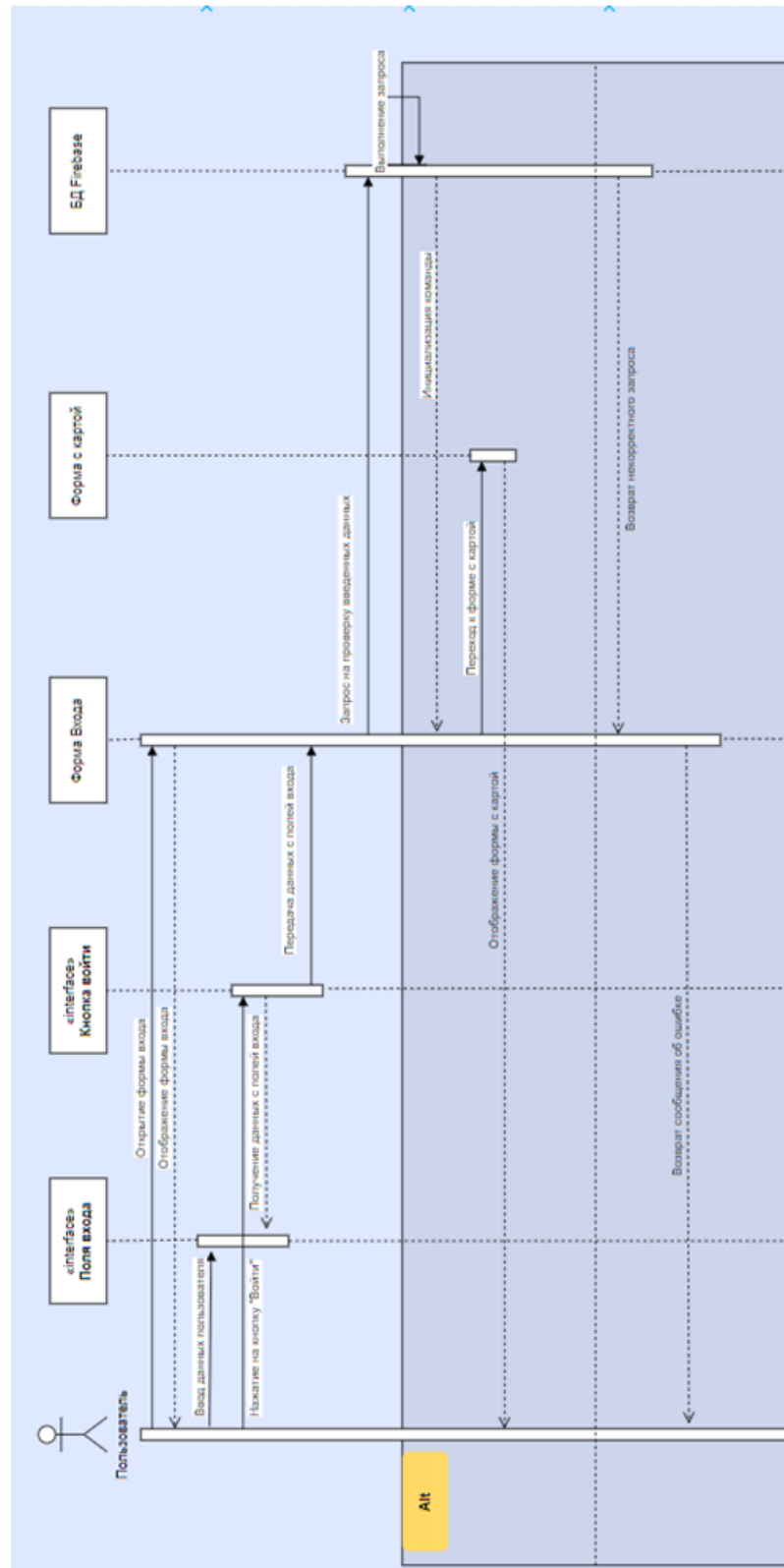


Рисунок В.1 — Диаграмма последовательности для варианта использования «Вход в приложение»

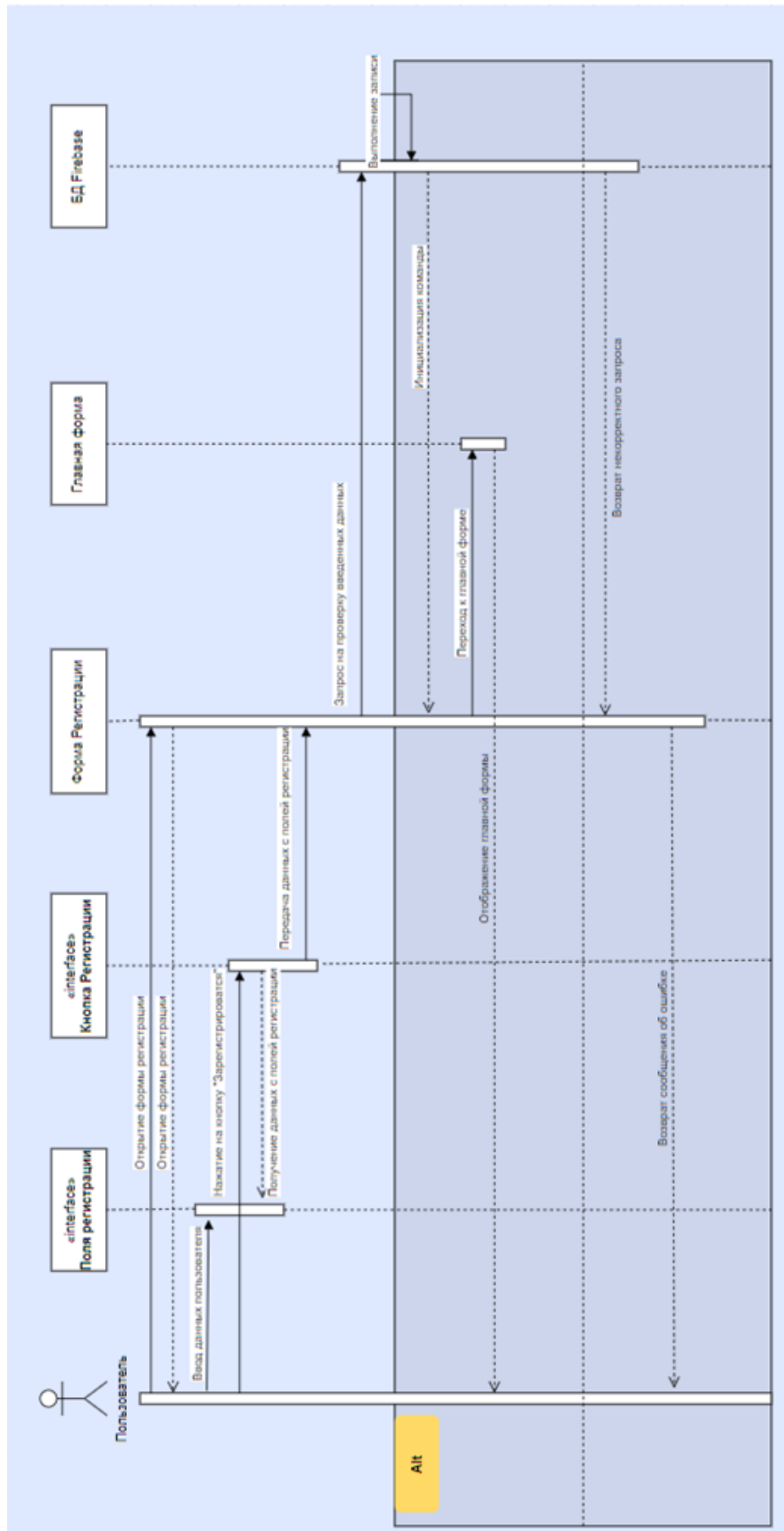


Рисунок В.2 — Диаграмма последовательности для варианта использования «Регистрация»