

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование кафедры)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Прикладная информатика в социальной сфере

(направленность (профиль)/специализация)

БАКАЛАВРСКАЯ РАБОТА

на тему: «Разработка геоинформационной системы лесоустройства г. Тольятти»

Студент

И.В. Борсук

(И.О. Фамилия)

(личная подпись)

Руководитель

Е.А. Ерофеева

(И.О. Фамилия)

(личная подпись)

Консультанты

М.А Четаева

(И.О. Фамилия)

(личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский

(ученая степень, звание, И.О. Фамилия)

(личная подпись)

« _____ » _____ 20 _____ г.

Тольятти 2019

АННОТАЦИЯ

Темой данной выпускной квалификационной работы является «Разработка геоинформационной системы лесоустройства г. Тольятти».

Целью данной выпускной квалификационной работы является разработка географической информационной системы.

Актуальность темы «Разработка географической информационной системы лесоустройства г. Тольятти» обуславливается потребностью в разработке географической информационной системы, которая позволит автоматизировать деятельность сотрудников лесничества г. Тольятти, а именно процесс учета лесного фонда.

Объектом исследования является процесс учета лесного фонда.

Предметом исследования является автоматизация процесса учета лесного фонда.

В первой главе проводится анализ деятельности лесничества г. Тольятти. Основной бизнес-процесс разбивается на этапы, после чего выделяются этапы, нуждающиеся в автоматизации. Производится сравнение и анализ представленных на рынке решений и ставится задача на разработку географической информационной системы.

Во второй главе производится выбор технологии логического моделирования, разрабатывается логическая модель географической информационной системы, производится проектирование базы данных.

В третьей главе описывается обоснование архитектуры проектируемой системы. Производится выбор технологии разработки серверной и клиентской части программного обеспечения, выбор системы управления базами данных, разрабатывается физическая модель данных. Описывается процесс разработки программной системы, описывается функциональность системы.

Выпускная квалификационная работа содержит пояснительную записку объемом 61 страницы, включая 23 иллюстраций, 4 таблицы, список литературы из 23 наименований, приложение.

ABSTRACT

The title of the bachelor's thesis is "Development of a geographic information system of Togliatti forestry".

The relevance of the bachelor's thesis is due to the need to automate the work of Togliatti forestry.

The aim of the work is to develop a geographic information system for automating the work of Togliatti forestry.

The object of the bachelor's thesis is the process of management of forest estate.

The subject of the bachelor's thesis is the automation of management of forest estate.

The first chapter is devoted to the analysis and functional modeling of domain processes. The main business process of the forest estate is divided into several stages and then analyzed. On the basis of the analysis, the task of developing the geographic information system is formulated.

The second chapter focuses on the choice of logical modeling technology, the development of a logical system model, and the logical database design.

The third chapter is mostly about the physical design of the information system. It includes the choice of information system architecture, the choice of development technologies, the choice of a database management system, and a description of the physical data model. The chapter also contains a description of the development process and the main functionality of the resulting system.

The bachelor's thesis consists of an explanatory note on 61 pages, including 23 figures, 4 tables, the list of 23 references, and an appendix.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
ГЛАВА 1 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ.....	8
1.1 Техничко-экономическая характеристика деятельности лесничества г. Тольятти.....	8
1.2 Концептуальное моделирование процесса учета лесного фонда	11
1.2.1 Выбор технологии концептуального моделирования процесса учета лесного фонда.....	11
1.2.2 Разработка и анализ модели бизнес-процесса учета лесного фонда «КАК ЕСТЬ».....	12
1.2.3 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии	16
1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям.....	17
1.4 Постановка задачи на разработку проекта создания географической информационной системы	18
1.5 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»	20
Выводы по главе 1.....	22
ГЛАВА 2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ГЕОИНФОРМАЦИОННОЙ СИСТЕМЫ.....	23
2.1 Выбор технологии логического моделирования географической информационной системы	23
2.2 Логическая модель географической информационной системы	24
2.3 Проектирование базы данных географической информационной системы.	30
2.3.1 Выбор технологии проектирования базы данных географической информационной системы	30
2.3.2 Разработка концептуальной модели данных географической информационной системы	31

2.3.3 Разработка логической модели данных АИС.....	32
2.4 Требования к аппаратно-программному обеспечению географической информационной системы	34
Выводы по главе 2.....	34
ГЛАВА 3. ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ГЕОГРАФИЧЕСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	35
3.1 Выбор архитектуры географической информационной системы.....	35
3.2 Выбор технологии разработки программного обеспечения географической информационной системы	36
3.2.1 Выбор серверной технологии	36
3.2.2 Выбор клиентской технологии	39
3.3 Выбор системы управления базой данных географической информационной системы	41
3.4 Разработка физической модели данных географической информационной системы.....	43
3.5 Разработка программного обеспечения географической информационной системы.....	44
3.6 Описание функциональности географической информационной системы	48
Выводы по главе 3.....	51
ЗАКЛЮЧЕНИЕ	52
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	53
ПРИЛОЖЕНИЕ А	56
ПРИЛОЖЕНИЕ Б.....	57

ВВЕДЕНИЕ

Информационные системы в настоящее время используются в различных сферах деятельности человека. Зачастую появляется необходимость определения пространственного положения объекта, или расчетов, связанных с географическим положением комплекса объектов. Для решения таких задач разрабатываются географические информационные системы. Географическая информационная система (ГИС) – это компьютерная система, для сбора, хранения и отображения географических данных. ГИС создаются для отображения различных видов данных на одной карте, это позволяет пользователям легко увидеть и проанализировать закономерности и отношения. Географические информационные системы нашли применение в самых разных областях человеческой деятельности, одной из которых является лесоустройство.

Одной из главных задач лесного хозяйства является хранение разносторонней информации и лесном фонде, разработка системы мероприятий, направленных на обеспечение рационального пользования лесным фондом, эффективного воспроизводства, охраны и защиты лесов. Одним из самых важных источников формирования информационных ресурсов о состоянии окружающей среды являются данные, получаемые лесничеством. Эти данные помогают рационально использовать природные ресурсы, выявлять потенциально опасные объекты.

Объектом исследования является процесс учета лесного фонда и ведение государственного лесного реестра.

Предмет исследования: автоматизация процесса учета лесного фонда.

Целью работы является разработка географической информационной системы.

Задачи:

- описать существующий бизнес-процесс учета лесного фонда;
- исследовать основной бизнес-процесс учета лесного фонда;

- выявить недостатки основного бизнес-процесса;
- проанализировать учебно-методическую и научную литературу, необходимую для разработки географической информационной системы;
- спроектировать географическую информационную систему;
- разработать географическую информационную систему.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка использованных источников и приложения.

В первой главе проводится анализ деятельности лесничества г. Тольятти. Основной бизнес-процесс – учет лесного хозяйства разбивается на этапы, после чего выделяются этапы, нуждающиеся в автоматизации. Производится анализ существующих разработок на предмет соответствия требованиям, сформулированным в соответствии с нуждами предприятия. На основе проведенного анализа ставится задача на разработку географической информационной системы.

Во второй главе разрабатывается логическая модель географической информационной системы, формируются требования к аппаратно-программному обеспечению системы.

В третьей главе описывается обоснование архитектуры проектируемой системы. Производится выбор технологии разработки серверной и клиентской части программного обеспечения, выбор системы управления базами данных, разрабатывается и описывается физическая модель данных. Описывается процесс разработки программного обеспечения, описывается функциональность системы.

ГЛАВА 1 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ

1.1 Техничко-экономическая характеристика деятельности лесничества г. Тольятти

В состав Тольяттинского лесничества в границах городского округа Тольятти включены леса Тольяттинского лесничества Министерства лесного хозяйства, охраны окружающей среды и природопользования Самарской области в соответствии с приказом Рослесхоза от 30.12.2008 № 435 «Об определении количества лесничеств на территории Самарской области и установлении их границ». Тольяттинское лесничество (далее – лесничество) расположено в северо-западной части Самарской области на территории городского округа Тольятти. Лесничество граничит: - на севере, востоке и западе – городской чертой г. Тольятти. - на юге – примыкает к приплотинному участку Куйбышевского водохранилища. Протяжённость лесничества с севера на юг- 11,3 км, с востока на запад 24,4 км.

Общая площадь лесничества по состоянию на 31.12.2013 г. составляет 7979 га. Площади участковых лесничеств: Васильевское участковое лесничество 2601 га, Тольяттинское участковое лесничество 5378 га.

Территория лесничества находится в бассейне реки Волга представленной участком Куйбышевского водохранилища. Лесничество расположено в северо-западной части Самарской области. Основная часть лесничества представлена компактным лесным участком, есть лесные участки, расположенные на островах Куйбышевского водохранилища. Количество лесных кварталов 111 шт. средняя площадь квартала - 72 га.

Целью деятельности лесничества является обеспечение реализации предусмотренных законодательством Российской Федерации полномочий министерства природных ресурсов, лесного хозяйства и экологии Новгородской области в сфере лесных отношений на подведомственной территории.

Для достижения указанных целей лесничество осуществляет следующие виды деятельности:

- ведение учета лесного хозяйства;
- осуществление мероприятий по использованию, охране, защите и воспроизводству лесов;
- осуществление федерального государственного лесного надзора (лесной охраны) на подведомственной территории;
- осуществление федерального государственного пожарного надзора в лесах на подведомственной территории.

Для дальнейшего исследования деятельности предприятия необходимо рассмотреть организационную структуру лесничества г. Тольятти (рисунок 1.1).

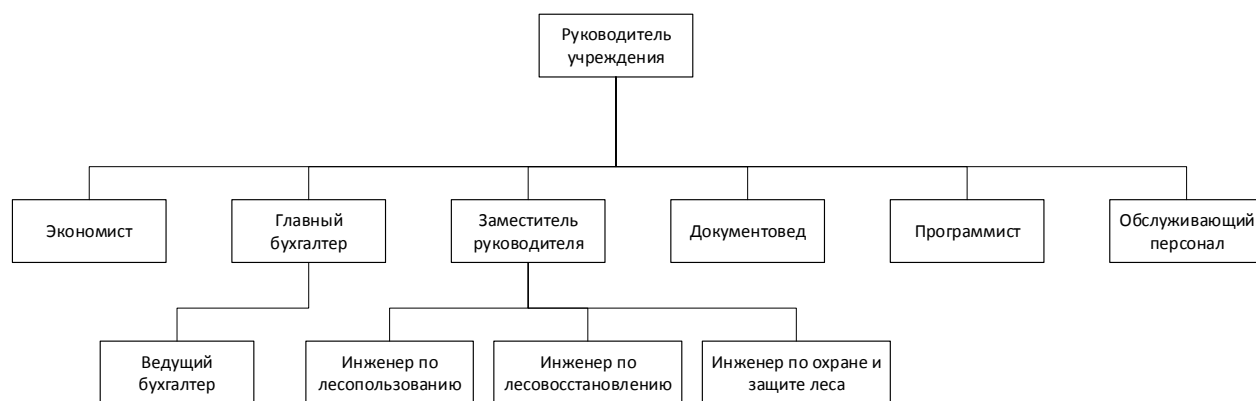


Рисунок 1.1 - Организационная структура предприятия

Главный бухгалтер обеспечивает организацию бухгалтерского учета, контроль и отражение на счетах бухгалтерского учета всех осуществляемых учреждением хозяйственных операций, предоставление оперативной информации, составление в установленные сроки бухгалтерской отчетности, осуществление совместно с другими подразделениями и службами экономического анализа финансово-хозяйственной деятельности и т.д.

Экономист осуществляет работу по экономическому планированию на предприятии, направленному на организацию рациональной хозяйственной деятельности, выявление и использование резервов производства с целью достижения наибольшей результативности деятельности предприятия и т.д.

Инженер по охране и защите леса занимается составлением планов ведения лесопатологических обследований и планов по проведению профилактических биотехнических мероприятий, принятием участия в приемке работ по проведению лесопатологических обследований, проверкой санитарно-оздоровительных мероприятий и т.д.

Инженер по лесопользованию занимается организацией мероприятий по повышению продуктивности лесов в соответствии с регламентом лесничества, организацией выполнения работ по видам лесопользования и т.д.

Инженер по лесовосстановлению организует работу по выращиванию посадочного материала, лесосеменному делу и лесовосстановлению, обеспечивает разработку проектов создания лесных культур и защитных лесонасаждений и подготовку их для утверждения, контролирует агротехнические сроки и качество выполнения лесокультурных работ, работ в питомниках по заготовке, переработке, хранению лесных семян и созданию постоянной лесосеменной базы и т. д.

Документовед разрабатывает и внедряет технологические процессы работы с документами и документной, принимает участие в планировании, организации и совершенствовании деятельности службы документационного обеспечения управления, осуществляет контроль за состоянием делопроизводства, принимает меры по упорядочению состава документов и информационных показателей, разрабатывает систему документации и т. д.

Программист разрабатывает на основе анализа математических моделей и алгоритмов решение экономических и других задач программы, обеспечивающие возможность выполнения алгоритма и соответственно поставленной задачи средствами вычислительной техники, проводит их тестирование и отладку.

К обслуживающему персоналу относятся водитель, сторож и уборщик.

Все должностные обязанности и организация работы сотрудников более подробно представлены в должностных инструкциях, согласованных с приказом руководителя лесничества г. Тольятти.

1.2 Концептуальное моделирование процесса учета лесного фонда

1.2.1 Выбор технологии концептуального моделирования процесса учета лесного фонда

Концептуальное моделирование предметной области является одним из ключевых этапов проектирования информационной системы. В настоящее время существует множество методологий концептуального моделирования информационных систем. Описание самых распространенных методологий приведено ниже.

IDEF0 – методология графического моделирования бизнес-процессов. Она предусматривает построение диаграмм как иерархической системы: сначала создается контекстная диаграмма, в которой описывается система в целом, потом строится диаграмма декомпозиции, где целая система разбивается на части-подсистемы, и каждая из них описывается по отдельности.

DFD – методология графического структурного анализа, описывающая потоки данных. Используя ее, можно показать последовательность работ, осуществляемых в ходе процесса, и потоки информации, которые передаются между данными работами. Другими словами, она дает возможность увидеть, как входные данные изменяются под действием каждого процесса в выходные.

BPMN – методология моделирования, анализа и реорганизации бизнес-процессов. Основной целью является обеспечение доступной нотацией описания бизнес-процессов каждого пользователя: от аналитиков и разработчиков, которые ответственные за введение технологий производства бизнес-процессов, до руководителей и простых пользователей, которые руководят данными бизнес-процессами и следят за их выполнением.

После рассмотрения самых популярных методологий моделирования в качестве методологии концептуального моделирования была выбрана методология IDEF0 по следующим причинам: IDEF0 представляет всю деятельность предприятия в нужном объеме, является легкой в изучении и позволяет детально рассмотреть процесс трансформации входных данных в выходные.

1.2.2 Разработка и анализ модели бизнес-процесса учета лесного фонда «КАК ЕСТЬ»

Для исследования деятельности предприятия были использованы следующие методы: сбор и анализ документов, интервьюирование инженеров лесного хозяйства.

Перед началом проектирования системы необходимо построить модель функционирования лесничества «AS-IS». Модель «AS-IS» («КАК ЕСТЬ») – это модель, описывающая основной бизнес-процесс предприятия, принципы и механизмы предприятия до ввода в эксплуатацию ЛГИС. Анализ бизнес-процесса учета лесного фонда выполнен с помощью IDEF0 диаграмм, так как IDEF0 является одной из самых наглядных методологий функционального моделирования. Разработанные диаграммы станут основой для комплексного анализа процессов, поиска проблем и путей их преодоления.

На рисунке 1.2 представлена модель бизнес-процесса учета лесного фонда «AS-IS», построенная по методологии IDEF0.

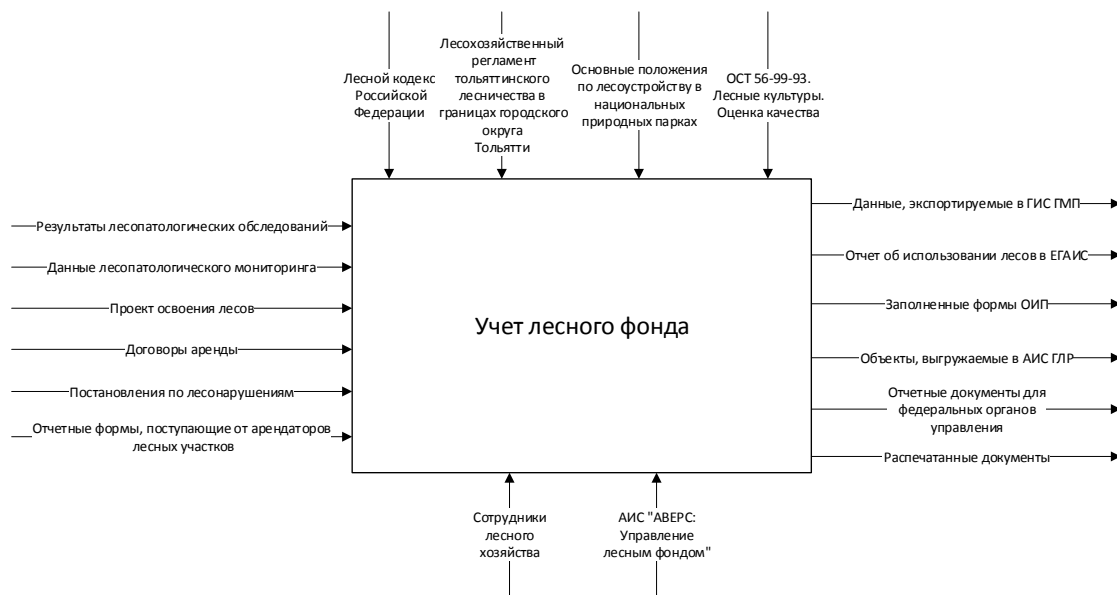


Рисунок 1.2 – Диаграмма IDEF0 (AS-IS) «Процесс учета лесного фонда»

Процесс учета лесного фонда является основным бизнес-процессом лесничества. Данный процесс получает на вход множество данных, таких как:

- результаты лесопатологических обследований и данные лесопатологического мониторинга;

- данные проекта освоения лесов;
- договоры аренды лесных участков;
- постановления по лесонарушениям;
- отчетные формы, поступающие от арендаторов лесных участков;

Далее процесс при помощи механизмов управления (стрелки с направлением сверху-вниз) и механизмов (стрелки с направлением снизу-вверх), выполняющих определенную работу, выдаёт следующий результат: отчет об использовании лесов в ЕГАИС, заполненные формы ОИП, объекты, выгружаемые в АИС ГЛР, отчетные документы для федеральных органов управления, распечатанные документы.

Для более подробного рассмотрения основного бизнес-процесса предприятия «учет лесного фонда» необходимо выполнить его декомпозицию (рисунок 1.3).

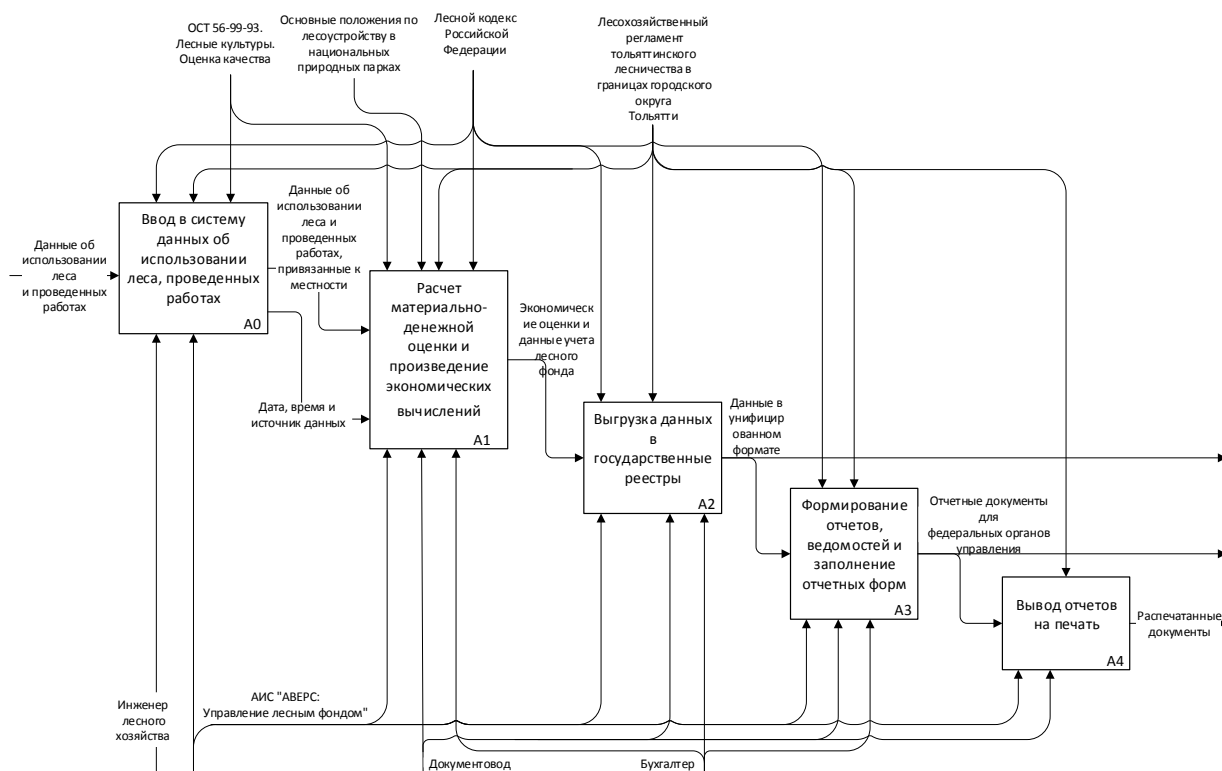


Рисунок 1.3 – Декомпозиция диаграммы IDEF0 (AS-IS) «Процесс учета лесного фонда»

На рисунке 1.3 показано, какие сотрудники участвуют на каждом этапе основного процесса. Для упрощения диаграммы входные данные были объединены в один входной поток «Данные об использовании леса и проведенных работах». Первый этап – «Ввод в систему данных об использовании леса, проведенных работах» производится инженером лесного хозяйства. При помощи АИС «АВЕРС» инженер вводит и привязывает к местности множество различных данных, таких как результаты лесопатологических обследований, данные лесопатологического мониторинга и т.д. (полный перечень данных приведен на рисунке 1.2). Далее документовед и бухгалтер при помощи АИС «АВЕРС» производят регламентированные экономические расчеты, такие как расчет материально-денежной оценки, руководствуясь лесным кодексом и другими отчетными документами. Затем происходит автоматическая выгрузка выбранных данных в государственные реестры, такие как государственный лесной реестр, государственная информационная система о государственных и муниципальных платежах и другие. Следующий этап – формирование отчетов, ведомостей, и заполнение отчетных форм и других отчетных документов, предоставляемых в федеральные органы управления. Заключительный этап – вывод отчетных документов на печать, в рамках которого некоторые документы выгружаются из системы в выбранных форматах и печатаются. Данный этап актуален лишь для тех документов, которые, согласно регламенту, должны предоставляться в вышестоящие органы управления в распечатанном виде.

В данной работе наибольший интерес представляет процесс ввода данных в систему и привязка этих данных к местности, поэтому необходимо рассмотреть данный процесс детальнее, для чего была выполнена его декомпозиция.

На рисунке 1.4 механизм «Инженер лесного хозяйства» был декомпозирован на три механизма: инженер по лесопользованию, инженер по лесовосстановлению, инженер по охране и защите леса в целях формирования более точного представления о том, кем являются пользователи существующей

АИС. Декомпозируемый процесс содержит в себе следующие бизнес-процессы: ввод данных об участках леса и информации о них, ввод результатов лесопатологического обследования, ввод данных о проведенных санитарно-оздоровительных мероприятиях, ввод данных о мероприятиях, связанных с воспроизводством лесов и лесоразведением, ввод информации о различных лесоустроительных мероприятиях, ввод в систему договоров аренды, а также регистрация протоколов о нарушениях.

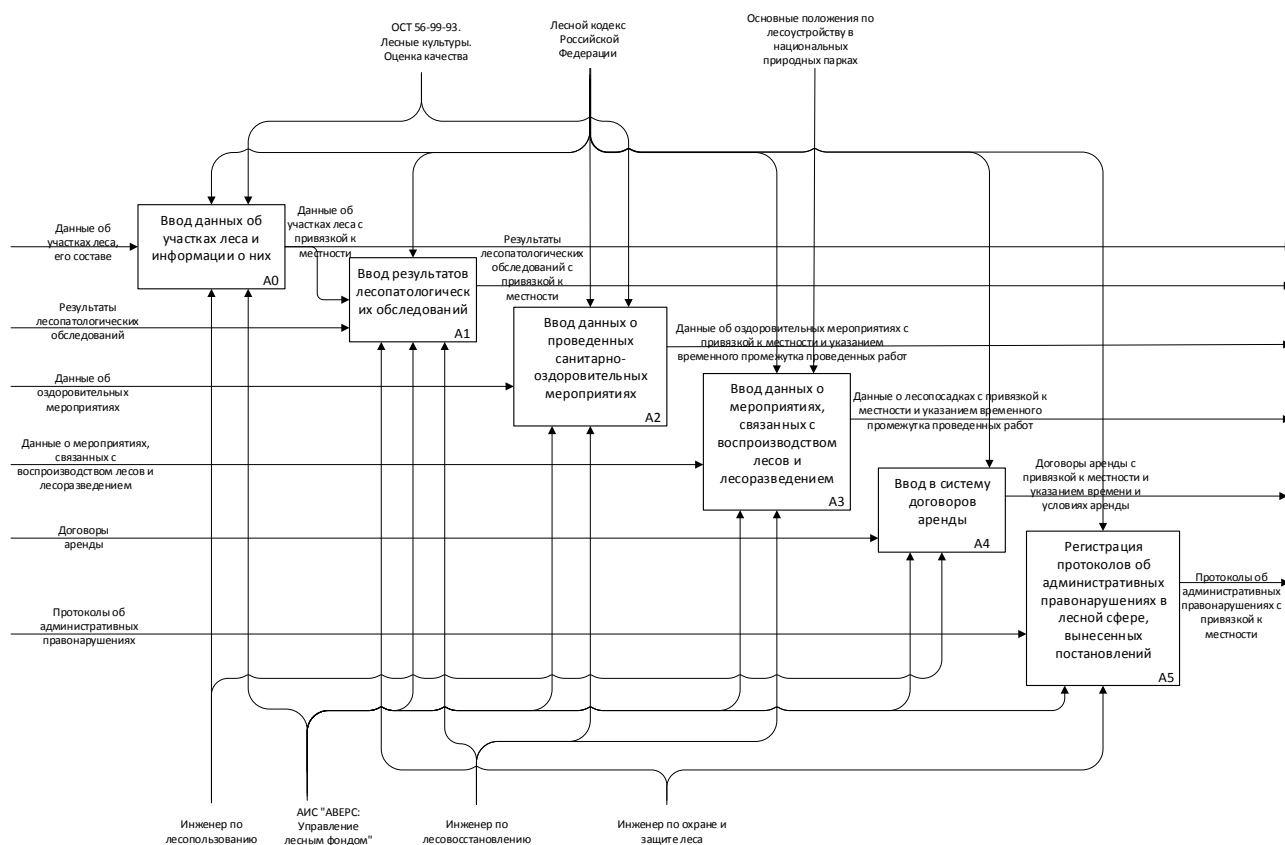


Рисунок 1.4 – Декомпозиция бизнес-процесса «Ввод в систему данных об использовании леса, проведенных работах» (AS-IS)

На основании вышеизложенного, можно прийти к выводу о том, что большая часть работы производится сотрудниками лесничества при помощи интерфейса АИС «АВЕРС», но существует ряд проблем, связанных с ручным выполнением операций ввода геодезических данных, отсутствием возможности ввода данных о патрулировании участков леса, отсутствием возможности ввода данных о нарушении с приложением фотографии. Также имеются проблемы с предоставлением данных лесничества в открытый доступ, отсутствием

интуитивно понятного интерфейса ввода и просмотра геодезических данных. Отсутствует возможность использования АИС на машинах без установленного клиента.

1.2.3 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии

На данный момент инженеры лесного хозяйства вынуждены вводить геодезические данные вручную, путем заполнения форм в АИС «АВЕРС». Данные формы являются неудобными и не привязываются к карте, что затрудняет процесс эксплуатации АИС. Также отсутствует возможность фиксирования данных о патрулировании на карте, что не позволяет оценить качество работ.

Используемая в настоящий момент система не позволяет неавторизованным пользователям просматривать информацию о состоянии леса, более того, просматривать информацию о лесе можно только установив клиент АИС «АВЕРС» на компьютер, что затрудняет использование системы.

Географическая информационная система может дать лесничеству следующие преимущества:

- удобный ввод данных в систему без установки клиента АИС;
- интуитивно понятный ввод данных при помощи интерактивной карты;
- контроль над качеством патрулирования;
- предоставление доступа к определенной информации всем желающим;
- ввод данных о нарушении с приложением фотографии;

Внедрение географической информационной системы в деятельность лесничества является немаловажным фактором, позволяющим добиться высоких результатов по контролю над лесопользованием и учету лесного фонда.

1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям

Для того, чтобы узнать, существует ли готовое решение, а также более точно определить дальнейшее направление проектирования, необходимо произвести анализ наиболее известных географических информационных систем. По итогам проведенного анализа необходимо узнать, является ли целесообразным решение о проектировании и разработке новой географической информационной системы.

В таблице 1.1 представлены требования, являющиеся критериями при анализе географических информационных систем.

Таблица 1.1 — Сравнительный анализ программных продуктов

Требование	Arcgis	QGIS	Mapinfo
Требования к функциональности системы			
Наличие веб-клиента	-	+	-
Наличие api для интеграции с бухгалтерской системой	-	-	-
Разграничение доступа	+	+	+
Наличие общедоступного режима	-	+	-
Поддержка формата геоданных mdb	+	-	+
Требования к удобству использования			
Наличие документации и руководства пользователя	+	+	+
Наличие интуитивно понятного интерфейса	-	-	-
Требование к производительности			
Скорость запуска не превышает 15 секунд	+	-	-
Время отклика системы при наличии большого количества объектов на карте не превышает 5 секунд	-	+	-

Продолжение таблицы 1.1

Требования к надежности			
Обеспечение одновременной работы 100 пользователей в режиме редактирования	-	-	-
Время восстановления системы после отказа не превышает 24 часа	+	-	+
Требования к поддержке и развитию			
Наличие бесплатной технической поддержки	-	+	-
Возможность расширения системы	-	-	-
Итого	5	6	4

В таблице 1.1 был проведен сравнительный анализ трех самых популярных программных продуктов. Подводя итоги данного сравнения, можно прийти к выводу, что нет ни одного программного решения, удовлетворяющего всем выдвинутым требованиям.

1.4 Постановка задачи на разработку проекта создания географической информационной системы

Для повышения продуктивности работы сотрудников тольяттинского лесничества, а также для осуществления общественного контроля необходимо разработать географическую информационную систему, удовлетворяющую всем требованиям предприятия.

Необходимо разработать географическую информационную систему с матричной моделью управления доступом. При запуске система будет доступна в режиме просмотра. Для внесения каких-либо изменений необходима авторизация. Данные учетной записи должны быть выданы лично каждому сотруднику.

Информация о лесных участках должна быть распределена по слоям, таким как «Использование лесов», содержащий информацию об инвестиционных проектах, нарушениях и т.д., «Лесоохрана», содержащий информацию о патрулировании и других проведенных работах, «Лесные пожары», содержащий информацию о времени и обширности пожаров, «Общественный контроль» для просмотра информации от людей, заметивших какое-либо нарушение, будь то незаконная рубка леса или несанкционированная свалка мусора.

В авторизованном режиме будет доступна панель инструментов, включающая в себя инструмент создания линий, многоугольников, прямоугольников, геопозиций, а также инструмент редактирования перечисленных выше географических объектов. По умолчанию авторизованный пользователь будет иметь доступ к абсолютно всем слоям данных. В режиме администратора система должна позволять разграничивать доступ: указывать, к каким слоям пользователь или группа пользователей имеет доступ.

В общедоступном режиме будет доступна информация о границах лесных участков, ответственных за патрулирование и проведение других лесоустроительных работ лицах. Также будет доступна информация, доступ к которой не запрещен согласно регламенту и информация, которая уже находится в открытом доступе в других государственных системах, таких как Росстат. К такой информации можно отнести: договоры аренды лесных участков, результаты лесопатологических обследований, данные о проведенных санитарно-оздоровительных мероприятиях, данных о патрулировании участков леса, данные о нарушениях.

Также в общедоступном режиме необходимо предусмотреть возможность ввода информации в слое «Общественный контроль» для регистрации незаконной рубка леса или несанкционированной свалки мусора.

1.5 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»

На основе построенной в пункте 1.2.2 модели «КАК ЕСТЬ» была построена модель «КАК ДОЛЖНО БЫТЬ», учитывающая изменения в рабочем процессе организации после внедрения географической информационной системы (рисунок 1.5).

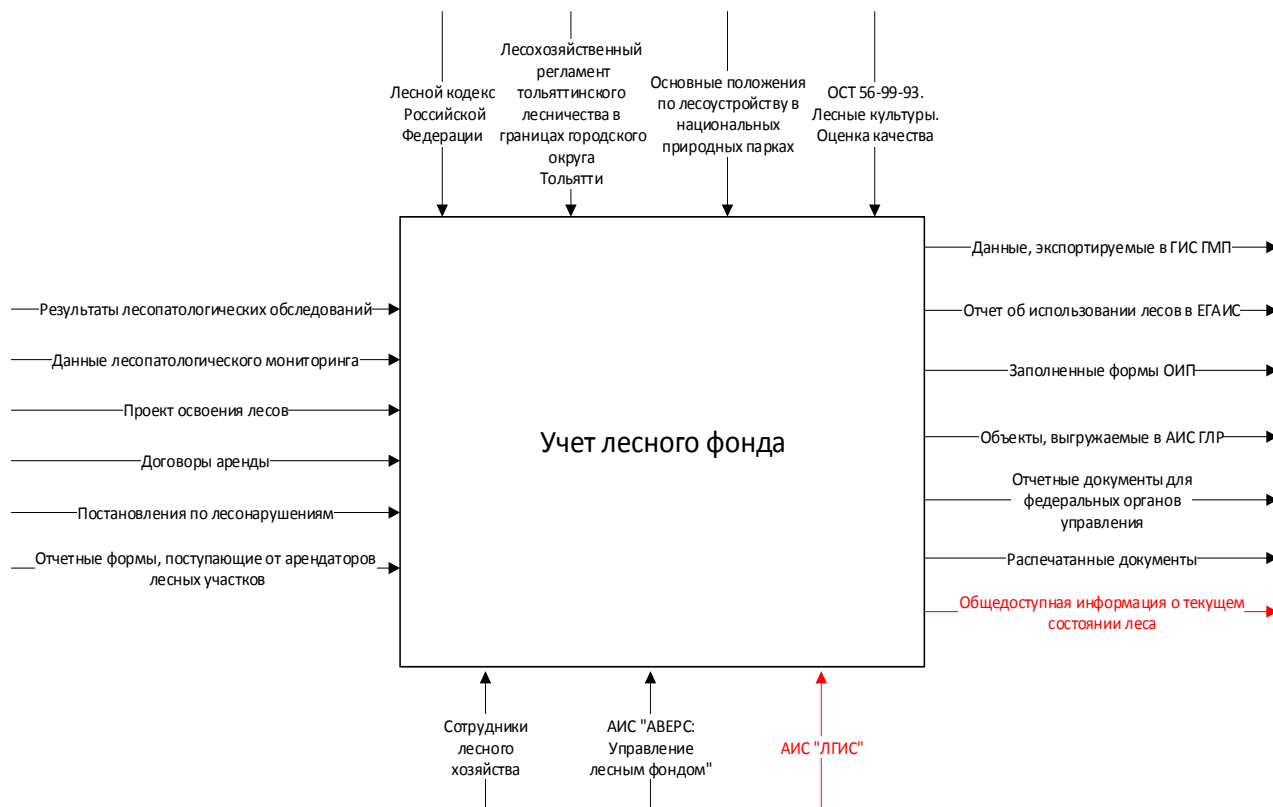


Рисунок 1.5 – Диаграмма IDEF0 («ТО-БЕ») «Процесс учета лесного фонда»

На рисунке 1.5 показано, что наряду с АИС «АВЕРС» к механизмам добавляется АИС «ЛГИС» (выделен красным цветом). В качестве результата данный процесс выдает общедоступную информацию о текущем состоянии леса. Декомпозиция бизнес-процесса «Учет лесного фонда» представлена на рисунке 1.6.

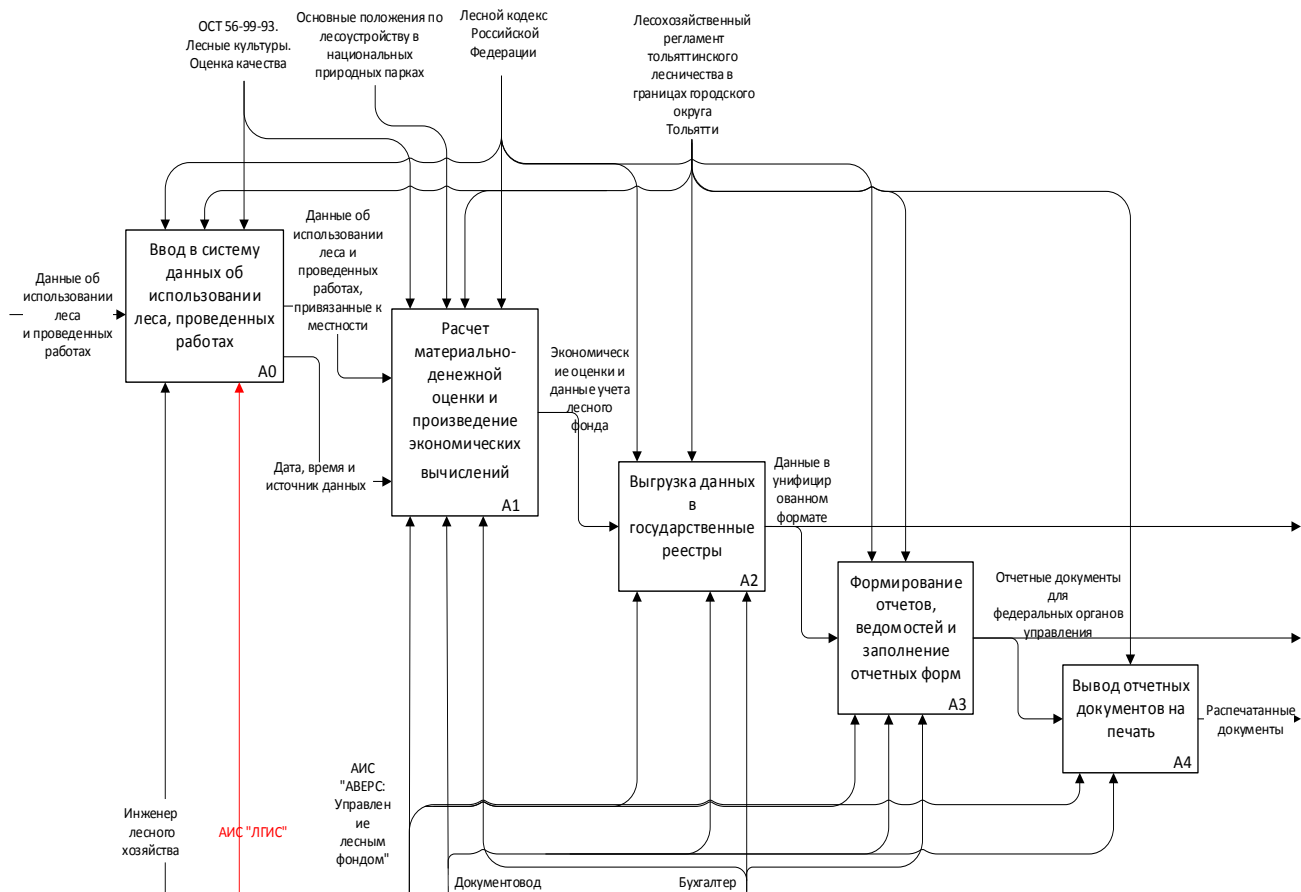


Рисунок 1.6 – Декомпозиция диаграммы IDEF0 («ТО-ВЕ») «Процесс учета лесного фонда»

На рисунке 1.6 изображена функциональная модель бизнес-процесса второго уровня, на которой отображен принцип работы с информацией: на первом этапе инженер лесного хозяйства вводит данные, пользуясь АИС «ЛГИС», на втором и последующих этапах сотрудники пользуются АИС «АВЕРС» для производства экономических вычислений, составления отчетов и выполнения других операций.

Для того, чтобы отобразить изменения в бизнес-процессе «Ввод в систему данных об использовании леса, проведенных работах» была произведена его декомпозиция (рисунок 1.7).

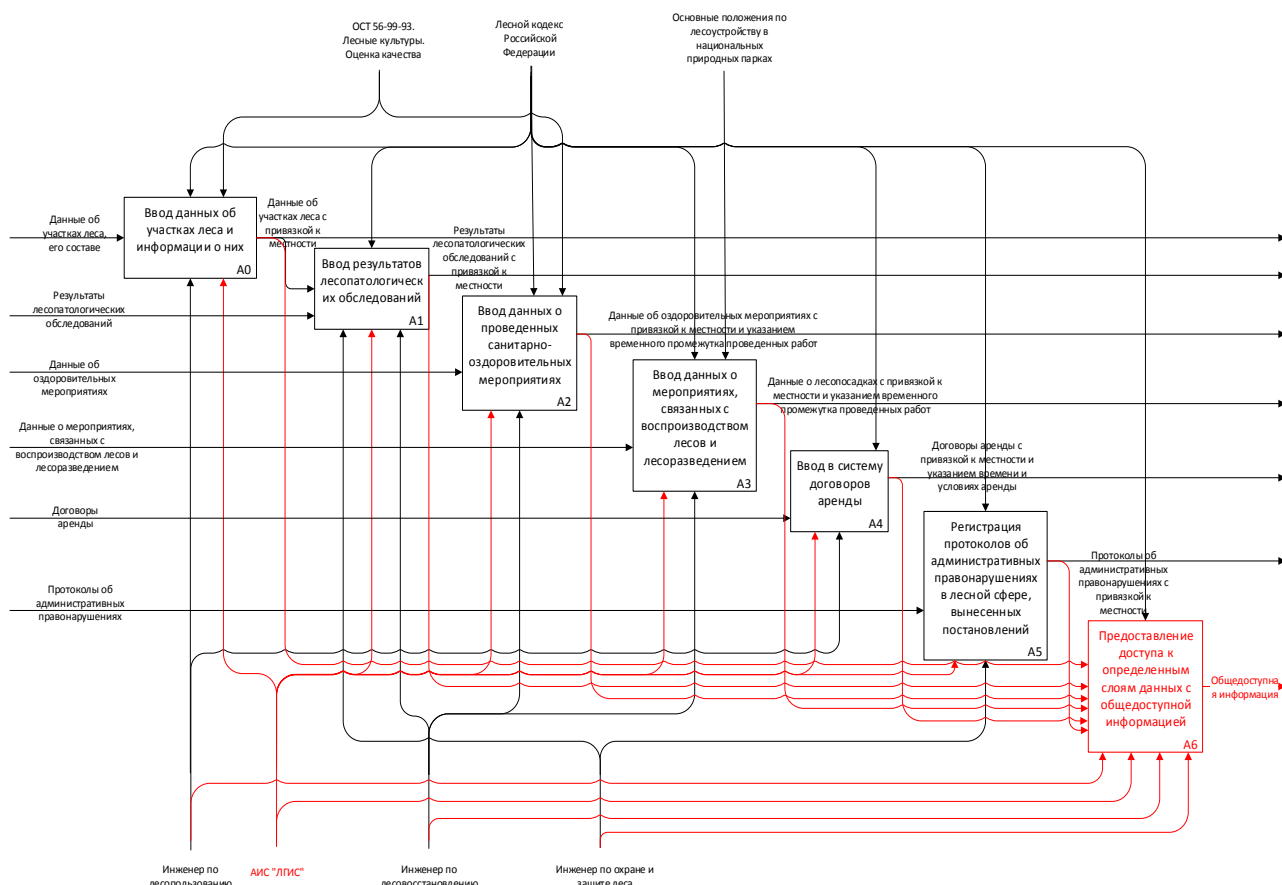


Рисунок 1.7 – Декомпозиция бизнес-процесса «Ввод в систему данных об использовании леса, проведенных работах» («ТО-ВЕ»)

На рисунке 1.7 более детально представлены изменения, касающиеся ввода данных в систему, а также изменения, касающиеся предоставления доступа к слоям данных.

Выводы по главе 1

Результатом анализа существующего бизнес-процесса учета лесного фонда были выявлены существенные недостатки, отрицательно сказывающиеся на эффективности труда сотрудников лесничества. Был предложен способ устранения выявленных недостатков путем внедрения географической информационной системы с интуитивно понятным интерфейсом.

Анализ известных решений показал, что они не соответствуют всем предъявленным к системе требованиям, вследствие чего было принято решение о проектировании и разработке географической информационной системы.

ГЛАВА 2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ГЕОИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Выбор технологии логического моделирования географической информационной системы

Логическое моделирование – это процесс разработки объектной модели системы и логической модели базы данных на основе созданной ранее концептуальной модели.

Логическое моделирование позиционируется как этап формализации модели проектируемой системы. Данный этап необходим для более точного представления выводов из концептуальной модели, а также постановки задачи на разработку программного обеспечения.

Согласно методологии бизнес-моделирования, на этапе построения логической модели будет применяться методология объектно-ориентированного анализа и проектирования, использующая нотацию UML.

Язык UML – унифицированный графический язык моделирования, другими словами, система обозначений, которая применяется для создания абстрактной модели разрабатываемой системы.

Для моделирования функций, выполняемых системой, будет построена диаграмма вариантов использования. Затем, для описания взаимодействия сотрудника лесничества с системой будет построена диаграмма деятельности.

Для проектирования структуры базы данных будет использоваться Case средство Egwin Data Modeler, с помощью которого будет разработана модель «сущность-связь» в нотации IDEFX. Для логического моделирования будет применяться объектно-ориентированная нотация языка UML пакета MS Visio.

Для демонстрации требований, описанных в первой главе и предъявляемых к системе, используется диаграмма вариантов использования. В диаграмме вариантов использования продемонстрирована совокупность прецедентов и акторов, а также существующие отношения между ними. С

помощью прецедентов моделируется поведение подсистемы или системы в целом.

Моделируя поведение элемента при помощи диаграммы вариантов использования, обеспечивается представление поведения информационной системы с высокой степенью детализации. Прецеденты делают возможным общение конечных пользователей и разработчиков на одном языке. Элементы, представляемые в диаграмме прецедентов, могут быть сложными образованиями с большим количеством операций и составных частей. Описание прецедентов определенного элемента, даст понимание конечным пользователям каким образом с ним обращаться.

2.2 Логическая модель географической информационной системы

Разработка логической модели географической информационной системы производится, в частности, для представления перехода от структурной диаграммы «КАК ДОЛЖНО БЫТЬ» к диаграмме вариантов использования (use case diagram), отражающей функциональный аспект логической модели системы.

Диаграмма вариантов использования необходима для следующих целей:

- определение контекста моделируемой предметной области на первом этапе проектирования системы;
- разработка общих требований к поведению системы;
- разработка первичной модели системы для последующей детализации.

Назначение диаграммы вариантов использования заключается в отражении функциональных возможностей системы, выявлении их взаимосвязи, выявления структуры системы.

Диаграмма вариантов использования строится при помощи ряда понятий, свойственных объектно-ориентированному подходу:

- исполнитель (актор) – личность или система, взаимодействующая с информационной системой. На диаграмме изображается в виде фигуры человека;
- прецедент – совокупность действий, инициированная объектом (личностью или системой), взаимодействующая с информационной системой. На диаграмме представляется в виде эллипса с надписью, характеризующей действие.
- ассоциация – связь между двумя объектами модели;
- обобщение – связь, служащая для того, чтобы показать, что одна сущность может быть обобщена до другой сущности;
- агрегация – связь, служащая для представления отношения типа «часть-целое» между целым и его частью.

Разработанная диаграмма прецедентов представлена на рисунке 2.1.

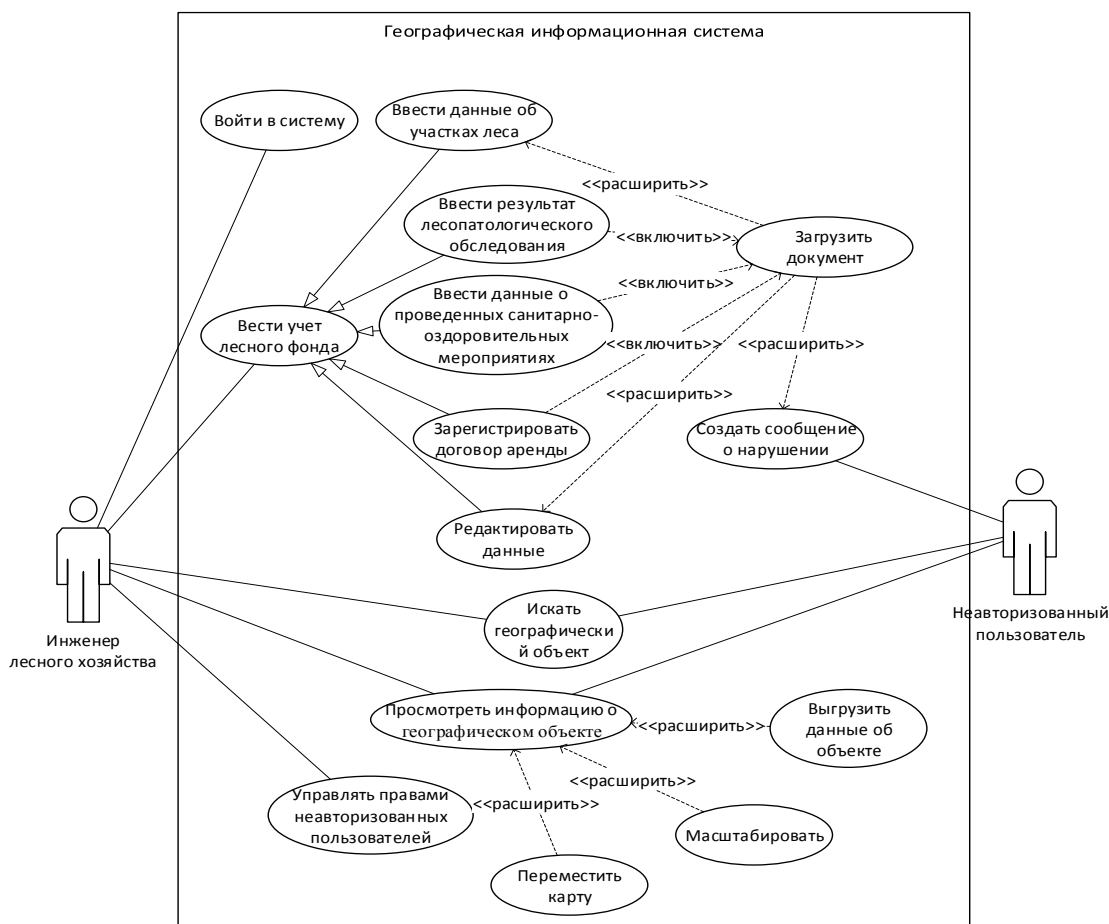


Рисунок 2.1 – Диаграмма вариантов использования географической информационной системы лесоустройства

В таблице 2.1 представлено краткое описание значимых прецедентов диаграммы, изображенной на рисунке 2.1.

Таблица 2.1 – Краткое описание прецедентов

Прецедент	Описание
Войти в систему	Пройти авторизацию (доступно сотрудникам лесничества)
Вести учет лесного фонда	Прецедент, обобщающий следующие прецеденты: ввести данные об участках леса, ввести результат лесопатологического обследования, редактировать данные и т.д.
Загрузить документ	Загрузить файл (документ, рисунок) для прикрепления к заполняемой форме
Искать географический объект	Произвести поиск по названию географического объекта
Просмотреть информацию о географическом объекте	Произвести манипуляции с картой, такие как перемещение, масштабирование, выделение кликом мыши для просмотра информации об интересующем объекте
Управлять правами неавторизованных пользователей	Назначать пользователям права для просмотра географических объектов
Создать сообщение о нарушении	Создать сообщение с описанием нарушения (незаконная вырубка леса, несанкционированная свалка мусора)

Разработанная модель вариантов использования позволила определить границы моделируемой предметной области, а также сформулировать общие требования к функциональному поведению проектируемой системы.

Для более детального рассмотрения поведения проектируемой системы необходимо разработать диаграммы деятельности. В диаграмме деятельности описывается последовательность шагов. Данный тип диаграмм подходит для отображения операций, которые пользователь производит, работая с системой.

Далее будет рассмотрен прецедент «Войти в систему»

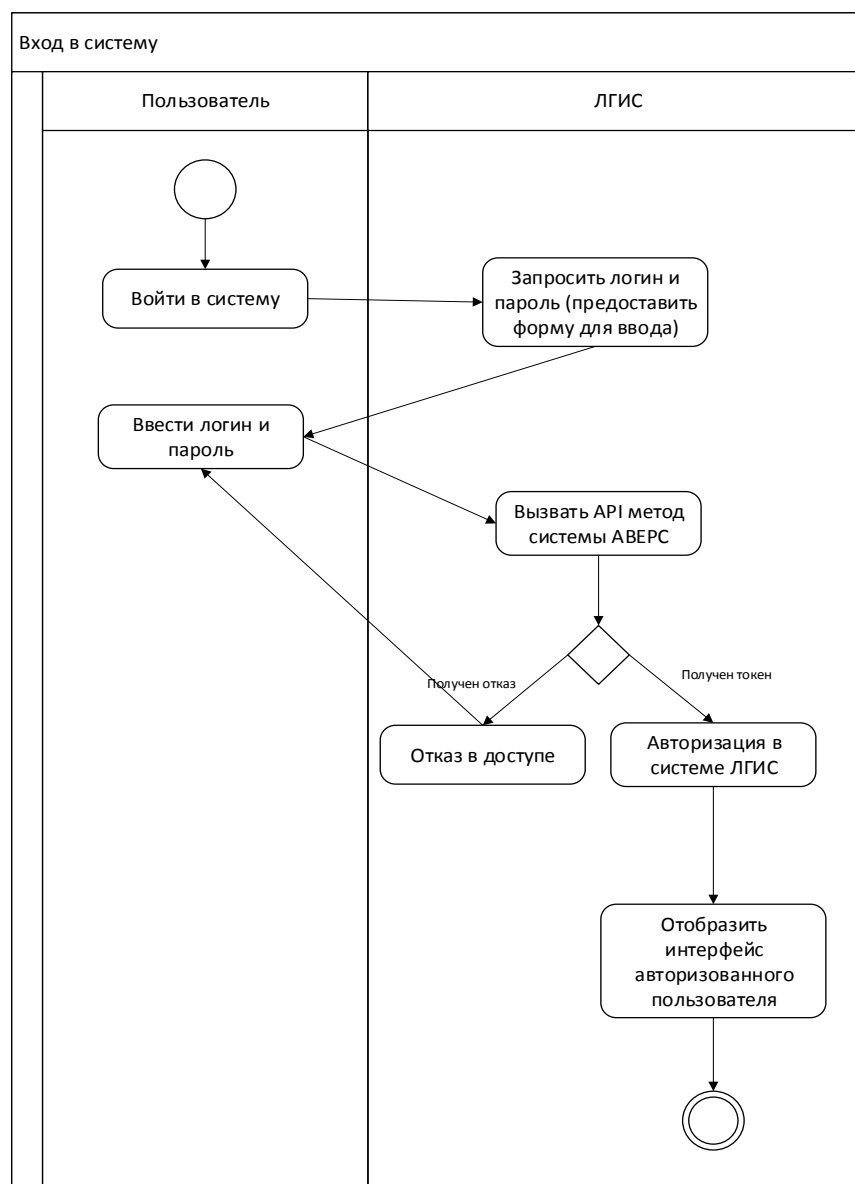


Рисунок 2.2 – Диаграмма деятельности «Войти в систему»

На рисунке 2.2 наглядно показано, какие процессы происходят в системе ЛГИС при входе в систему: система предоставляет пользователю форму для ввода логина и пароля, после чего введенные данные передаются в систему АВЕРС для генерации токена. В случае получения отказа пользователь остается на той же странице. В случае успешной аутентификации система отображает интерфейс авторизованного пользователя, включающий в себя инструменты редактирования.

Далее необходимо рассмотреть прецедент «Выгрузка данных» (рисунок 2.3).

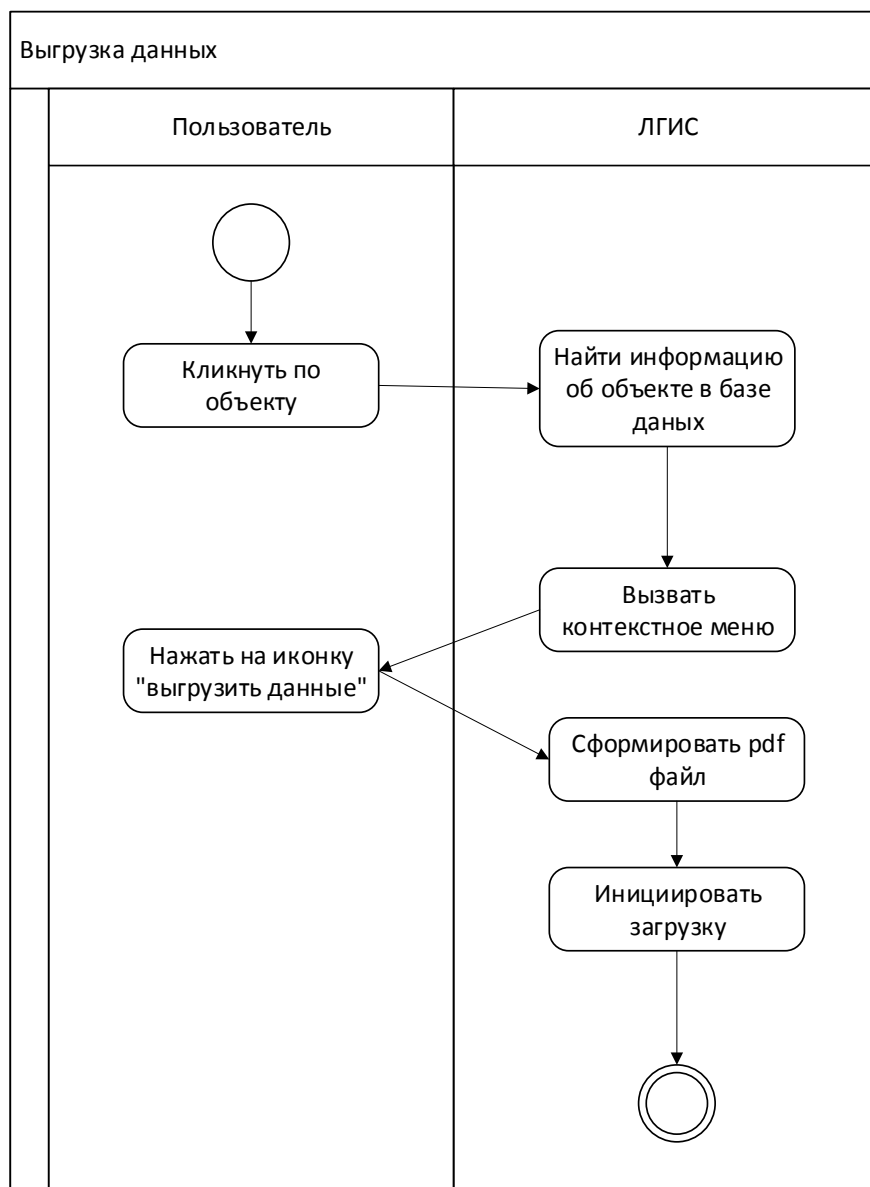


Рисунок 2.3 – Диаграмма деятельности «Выгрузить данные об объекте»

На рисунке 2.3 представлена последовательность действий пользователя для получения файла с данными о выбранном объекте в формате pdf и процессы, происходящие в системе при выполнении данной операции. После того, как пользователь кликнул по интересующему его объекту, система ЛГИС находит информацию об объекте в базе данных, после чего вызывает контекстное меню, содержащее информацию об объекте, и имеющее кнопку для выгрузки данных. После того, как пользователь нажимает на кнопку «выгрузить данные», система формирует pdf файл и иницирует загрузку файла.

Диаграммы классов используются при моделировании проектируемых систем наиболее часто. Они являются одной из форм статического описания системы с точки зрения ее проектирования, показывают ее структуру.

На рисунке 2.4 представлена диаграмма классов географической информационной системы.

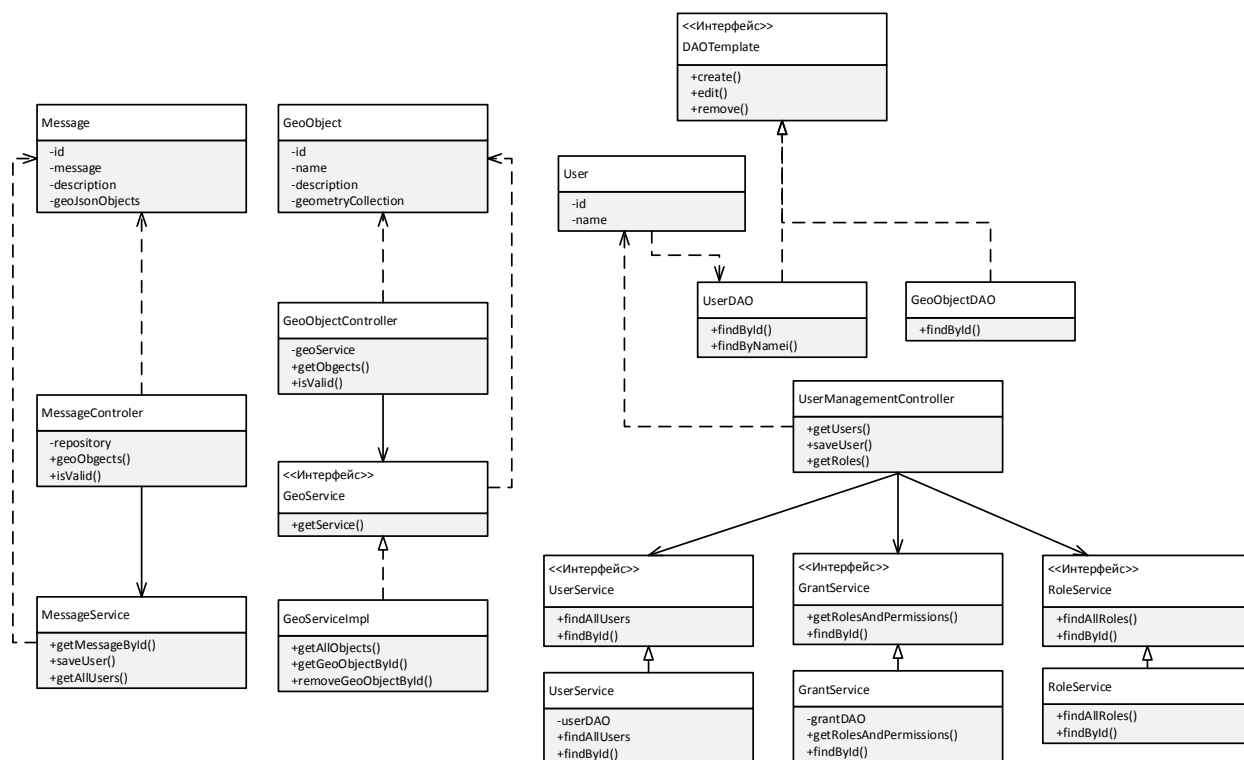


Рисунок 2.4 – Диаграмма классов географической информационной системы

На диаграмме, изображенной на рисунке 2.4 представлены основные сущности системы. Интерфейс GeoObject представляет собой верхний уровень иерархии классов системы. Класс GeoObjectController относится к слою представления. Данному классу передается запрос клиента, после чего контроллер обрабатывает запрос, создает модель и работает с сервисом. Интерфейс GeoService и его реализация – класс GeoServiceImpl отвечают за предоставление географических данных клиенту. UserDAO – это интерфейс с описанием общих методов, которые будут использоваться при взаимодействии с базой данных. В данном случае это методы поиска, удаление по ключу, обновление и т.д.

2.3 Проектирование базы данных географической информационной системы

2.3.1 Выбор технологии проектирования базы данных географической информационной системы

База данных представляет собой информационную модель предметной области, позволяющую хранить данные о группе объектов в упорядоченном виде. Любой объект реального мира имеет ряд характерных свойств. Свойства объектов отображаются в базе данных с помощью переменных величин.

В качестве способа организации данных была выбрана реляционная база данных, так как данный способ хранения данных является самым подходящим для проектируемой системы по следующим причинам:

- наглядность модели: данные в реляционной модели представлены в табличной форме;
- независимость данных: изменения структуры реляционной базы данных как правило не приводят к серьезным изменениям в прикладных программах;
- распространенность реляционных баз данных среди разработчиков: использование данного типа баз данных позволяет сэкономить бюджет и время.

Реляционная база данных – это набор таблиц, состоящих из столбцов и строк. Строка в базе данных содержит одну запись. Столбец содержит все экземпляры конкретного фрагмента данных всех строк. Как правило, каждая таблица имеет столбец с уникальным идентификатором.

Географическая информационная система основана на реляционной базе данных. Проектирование информационной системы будет выполнено с использованием модели «сущность-связь» (Entity Relationship). Данная модель представляет собой визуальное средство изображения объектов предметной области, их характеристик и отношений между объектами.

На данном этапе необходимо разработать модель базы данных на логическом уровне. Логический уровень представляет собой абстрактную модель

данных, на которой данные представляются так, как выглядят в реальном мире. Логический уровень не подразумевает описание конкретной СУБД, так же как не подразумевает определение типов данных.

Для проектирования реляционной базы данных будет применяться case-средство ERwin Data Modeler. Нотация IDEF1X будет использоваться как на логическом уровне, так и на физическом.

2.3.2 Разработка концептуальной модели данных географической информационной системы

Концептуальная модель базы данных подразумевает описание основных сущностей и отношений между ними. Данный тип модели является отражением предметной области, в рамках которого будет выполняться проектирование базы данных.

Для разработки концептуальной модели системы необходимо выделить следующие информационные объекты:

- географический объект;
- тип географического объекта;
- координаты;
- атрибут географического объекта;
- значение атрибута;
- документ;
- пользователь;
- права пользователя.

Данные объекты были выделены именно таким образом для обеспечения большой гибкости при создании новых данных. Объекты «географический объект», «атрибут географического объекта» и «значение атрибута» являются частью модели данных entity-attribute-value (EAV). Это означает, что при создании нового типа географического объекта не будет никакой необходимости создавать новые таблицы и поля в базе данных. Для более

детального рассмотрения необходимо перейти к разработке модели базы данных.

Все информационные объекты проектируемой базы данных представлены на диаграмме концептуальной модели данных (рисунок 2.5).

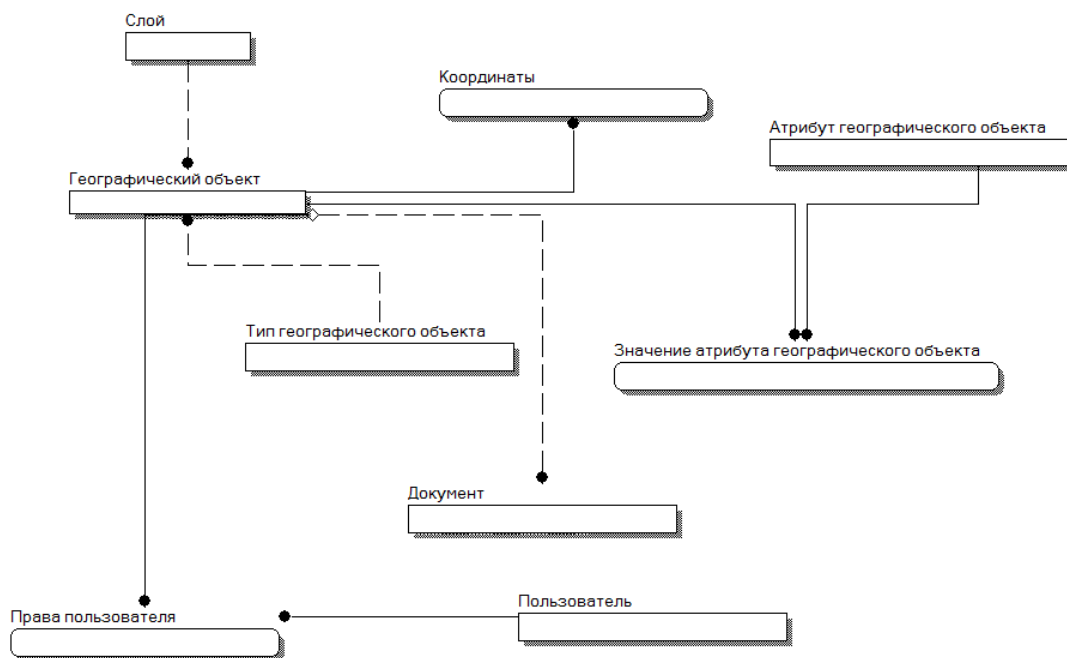


Рисунок 2.5 – Концептуальная модель базы данных географической информационной системы

На рисунке 2.5 изображена концептуальная модель базы данных, полностью соответствующая теоретическим требованиям. На рисунке представлены сущности, связи между сущностями, а также типы связей.

2.3.3 Разработка логической модели данных АИС

Логическая модель призвана расширить концептуальную модель путём определения для сущностей их атрибутов и ограничений. Логическая модель уточняет состав сущностей, а также взаимосвязи между ними. Таким образом, концептуальная модель изменяется так, чтобы она могла быть обеспечена конкретной моделью данных, в результате чего формируется логическая модель.

Основное назначение логической модели – отражение логических связей между элементами данных без зависимости от содержания этих элементов и среды их хранения.

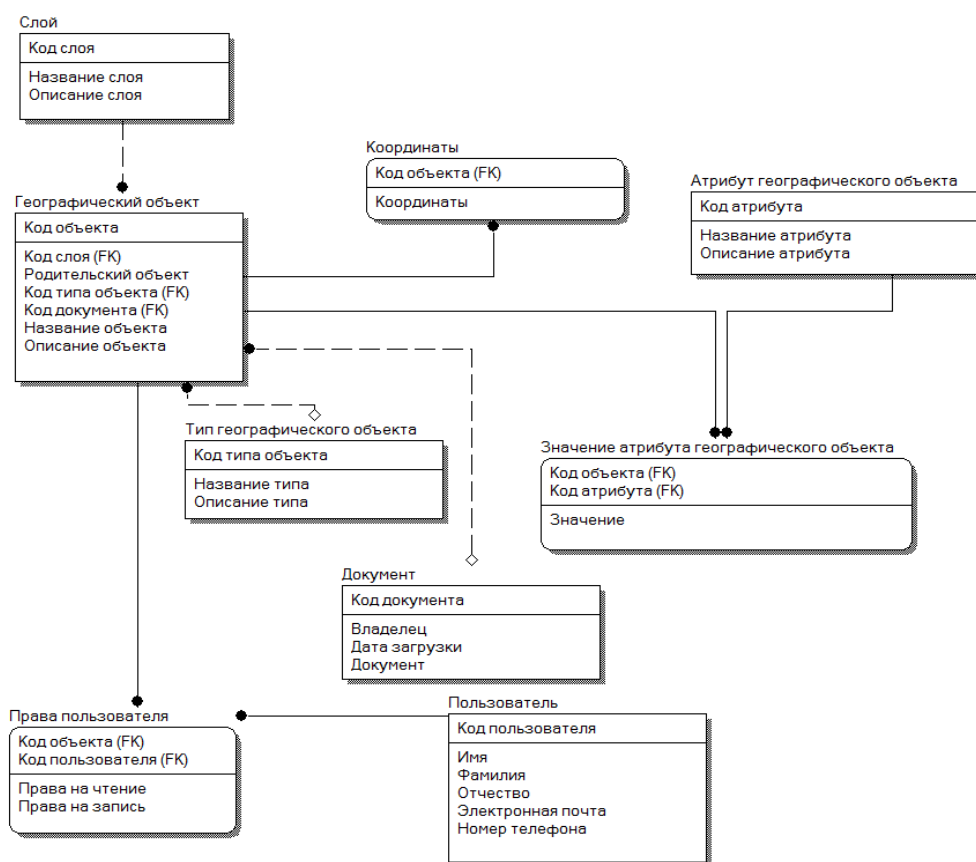


Рисунок 2.6 – Логическая модель базы данных географической информационной системы

На диаграмме, изображенной на рисунке 2.6 показано, что база данных состоит из таблиц «Географический объект», «Тип географического объекта», «Координаты», «Слой», «Атрибут географического объекта», «Значение географического объекта», «Документ», «Пользователь» и «Права пользователя».

Сущность «Географический объект» содержит атрибут «Код объекта», являющийся первичным ключом. Так как «Географический объект» выступает в роли центральной сущности модели данных EAV, у него имеется множество внешних ключей: «Код слоя», «Родительский объект», «Код типа объекта»,

«Код документа». Также имеются атрибуты «Название объекта» и «Описание объекта».

Сущность «Атрибут географического объекта» представляет собой хранилище названий атрибутов. Сущность «Значение географического объекта» представляет собой хранилище значений атрибутов, связанных с конкретным объектом. Сущность «Слой» содержит информацию о слое данных, к которому принадлежит тот или иной географический объект. Сущность «Тип географического объекта» содержит информацию о типе объекта. Сущность «Координаты» содержит информацию о местоположении объекта и его геометрии. Сущность «Документ» содержит в себе все документы, такие как акты, отчеты, постановления и т.д. Сущности «Пользователь» и «Права пользователей» необходимы для хранения необходимой информации о пользователях системы и их правах.

В данном параграфе было выполнено построение логической модели данных для сущностей, которые характеризую предметную область и позволяют полностью описать объект автоматизации. Далее необходимо определить требования к аппаратно-программному обеспечению географической информационной системы.

Выводы по главе 2

В данной главе было произведено логическое моделирование географической информационной системы: были разработаны диаграмма вариантов использования, диаграмма классов и диаграммы деятельности. Далее было произведено проектирование базы данных географической информационной системы: были определены ключевые сущности, установлены связи между ними, затем были описаны атрибуты сущностей. Затем было произведено описание требований к аппаратно-программному обеспечению.

ГЛАВА 3. ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ГЕОГРАФИЧЕСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

3.1 Выбор архитектуры географической информационной системы

Выбор архитектуры является неотъемлемой частью разработки АИС. В разрабатываемой географической информационной системе будет использована классическая трехзвенная архитектура клиент-сервер. Данная архитектура подразумевает наличие трёх компонентов. Компоненты и их физическое размещение изображены на диаграмме размещения (рисунок 3.1).

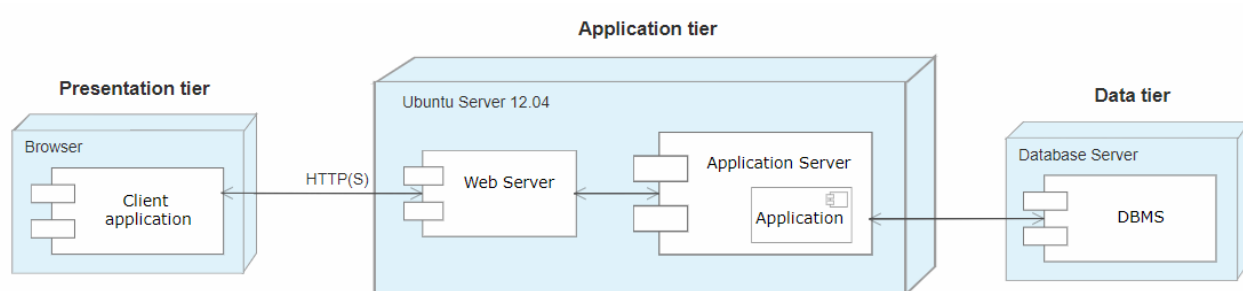


Рисунок 3.1 – Диаграмма размещения

Разделение компонентов приложения на разные уровни увеличивает надёжность и масштабируемость приложения. Это осуществляется за счет возможности внесения изменений в какой-либо уровень без необходимости изменять всё решение.

Уровень представления (**Presentation tier**) отвечает за взаимодействие пользователя с системой, содержит часть бизнес-логики приложения и осуществляет взаимодействие пользователя с приложением.

Уровень бизнес-логики (**Application tier**) координирует систему, обрабатывает команды, выполняет расчеты, направляет клиентское приложение к конкретному рабочему процессу, который будет его обслуживать.

Уровень доступа к данным предоставляет доступ к данным приложения, осуществляет взаимодействие с ним осуществляет уровень бизнес-логики.

Выбранная архитектура, изображенная на рисунке 3.1, обладает гибкостью, масштабируемостью, конфигурируемостью и безопасностью, что

делает ее наиболее подходящей для поставленных в рамках разработки системы задач.

3.2 Выбор технологии разработки программного обеспечения географической информационной системы

Для определения оптимальной технологии разработки необходимо рассмотреть наиболее часто используемые языки и фреймворки.

Для начала необходимо определить критерии выбора:

- оптимальность использования при средней сложности проекта;
- гибкость решения;
- наличие широкого сообщества;
- наличие подробной документации;
- возможность интеграции с другими решениями.

Выбор технологии по представленным выше критериям позволит добиться объективности выбора, что в дальнейшем позволит сэкономить время и деньги.

3.2.1 Выбор серверной технологии

Далее необходимо сравнить серверные технологии по существенным показателям.

Язык PHP имеет большую базу всевозможных решений от систем управления контентом до фреймворков. PHP является переносимым решением. Данный язык не зависит от конкретной платформы, он может быть запущен почти на любом сервере и на любой платформе. Интеграция PHP проектов и их развертывание можно произвести за короткий промежуток времени. Также такой подход позволяет запускать и управлять приложениями без знаний консольных команд. В отличие от Java, C# и других языков программирования общего назначения, PHP был разработан специально для создания веб-страниц.

Именно поэтому рассматриваемый язык содержит всю необходимую функциональность для работы с HTML, серверами баз данных.

К недостаткам PHP следует отнести плохое разделение ответственности. PHP не подходит для реализации паттерна MVC, который рекомендуется для использования в веб-разработке. Также минусом данного языка является то, что смешение кода HTML и PHP приводит к не очень красивому и сложно поддерживаемому коду, где бизнес-логика смешана с представлением.

Еще одним недостатком PHP является устаревшая клиент-серверная модель. PHP следует классической клиент-серверной модели где запрос страницы инициирует приложение, соединение с базой данных, их обработкой и рендерингом HTML. Это делает приложения на PHP медленными в сравнении с Node.js приложениями, которые инициируются при запуске. Именно поэтому Node.js больше подходит для написания приложений реального времени.

Технология Node.js позволяет создавать приложения с неблокирующим вводом/выводом, которые способны обрабатывать несколько запросов одновременно. Используя встроенную в JavaScript асинхронность, можно создавать хорошо масштабируемые серверные приложения, которые максимизируют использование одного ядра процессора при одновременной обработке большого количества запросов, чем при использовании обычных многопоточных серверов. Такая функциональность делает Node.js хорошим выбором для приложений реального времени, требующих большого количества операций ввода/вывода.

Большим плюсом использования данного языка является то, что и на клиенте, и на сервере будет использован один и тот же язык JavaScript. Существует множество фреймворков, таких как Angular, React или Vue, написанных на JavaScript, являющимся основным языком всех современных браузеров. Использование Node.js на сервере позволяет использовать все преимущества скриптового языка на обеих платформах. Еще одним плюсом данной технологии является отсутствие строгих правил и жестких

зависимостей, что открывает простор для творчества при разработке приложений.

К недостаткам Node.js следует отнести малую эффективность в операциях, интенсивно использующих ресурсы процессора. Событийно-ориентированная архитектура Node.js имеет некоторые ограничения, а именно – низкую эффективность при большой нагрузке на процессор. Несмотря на то, что Node.js хорошо справляется с конкурентной обработкой множества запросов, он все же плохо справляется с операциями вроде генерирования графических данных, обработки изображений.

Язык C# является объектно-ориентированным языком со строгой типизацией данных, что является большим преимуществом по сравнению с PHP и JavaScript. Данный язык поддерживает полиморфизм, позволяет перегружать операторы, позволяет работать с событиями, атрибутами, свойствами, делегатами, обобщенными типами, анонимными функциями с поддержкой замыканий.

К минусам языка C# следует отнести ориентированность, в основном, только на .NET платформу и необходимость покупки лицензии.

Java – это объектно-ориентированный язык программирования, обладающий C-подобным синтаксисом. Главное преимущество Java заключается в том, что при компиляции код транслируется в байт-код, поэтому приложения, написанные на данном языке, могут работать на любой платформе, на которой установлена виртуальная Java-машина (JVM). Гибкая система безопасности, в рамках которой выполнение программы полностью контролируется JVM, является важной особенностью технологии Java. Единственным большим недостатком Java является снижение производительности: для некоторых задач время выполнения байт-кода в полтора-два раза превышает время выполнения тех же операций с использованием C/C++.

Для наглядного сравнения серверных языков разработки была составлена таблица 3.1.

Таблица 3.1 – Сравнение языков программирования

Требование	PHP	JavaScript (Node.js)	C#	Java
Оптимальность использования при средней сложности проекта (0-3)	2	3	2	1
Гибкость решения (0-5)	2	4	3	5
Размер сообщества (0-3)	3	2	2	3
Наличие подробной документации (0-1)	1	1	1	1
Опыт работы с языком (0-5)	2	3	2	5
Итого	10	13	10	15

После проведения сравнительного анализа существующих языков, используемых при создании веб-проектов, были выбраны 4 кандидата: PHP, C#, Java, Javascript. Изучив существующую документацию языков и сравнив их слабые и сильные стороны сделан вывод, что наиболее подходящим языком для разработки географической информационной системы является Java. В связи с тем, что стек технологий Jakarta EE (Java EE) имеет большой порог вхождения, имеет тенденцию к уменьшению размера сообщества, в качестве фреймворка будет использоваться Spring Framework. Spring Framework представляет собой универсальный фреймворк с открытым исходным кодом для Java-платформы, обеспечивающий комплексную модель разработки и конфигурации для современных бизнес-приложений на Java.

3.2.2 Выбор клиентской технологии

Далее необходимо определиться с выбором клиентской технологии разработки, а именно – с выбором javascript фреймворка. Для анализа были

выбраны следующие фреймворки: Angular, React и Vue. Для начала необходимо произвести сравнение фреймворков по фактическим показателям, которые критичны для разработчика.

В первую очередь javascript фреймворки необходимо сравнить по рендерингу (отображению конечного результата). Существует два вида рендеринга: на стороне сервера или на стороне клиента, когда страница отрисовывается за счет мощностей машины пользователя.

DOM (Document Object Model) – это объектная модель документа, позволяющая считывать и изменять содержимое страницы, оформление и структуру HTML документов. Каждый из рассматриваемых фреймворков по своему обрабатывает DOM, что оказывает влияние на рендеринг конечной страницы, отображаемой на экране пользователя.

Vue.js и React создают копию DOM, обрабатывают ее, а затем результат сравнивается с исходной версией. В конечном документе (то есть на экране пользователя) заменяются только те части страницы, которые отличаются от результатов обработки. Такой подход значительно ускоряет загрузку и рендеринг страницы. Также данный подход сокращает объем трафика.

Подход к обработке DOM фреймворка Angular сильно отличается: рендеринг разделяется на два потока, за рендеринг DOM отвечает браузер, а за создание директив, загрузку кода и сервисов отвечает общий поток. Данный подход несомненно является минусом фреймворка, однако именно при таком подходе поисковая оптимизация не вызовет затруднений за счет того, что поисковым системам при индексации будет предоставлена корректная страница.

Что касается архитектуры компонентов приложений, то однозначно лучшими являются Vue и Angular, потому как применение React подразумевает необходимость поиска и внедрения дополнительных библиотек для реализации каждой из задач, необходимость настройки функциональной части приложения под конкретную библиотеку. В случае использования готовых фреймворков – Vue.js и Angular, проблем с подбором или настройкой библиотек для разных

задач уже не возникает. Высокоуровневый API обеспечивает обратную совместимость для всех библиотек. Это позволит подключиться к проекту стороннему программисту без длительного изучения архитектуры приложения. Именно в унификации процессов кроется популярность полноценных фреймворков.

Подводя итог вышесказанному, можно прийти к выводу о том, что Angular является самым разумным выбором для разработки географической информационной системы.

3.3 Выбор системы управления базой данных географической информационной системы

Выбор системы управления базой данных (СУБД) – один из важных этапов разработки информационной системы. СУБД должна соответствовать следующим требованиям:

- масштабируемость: СУБД должна учитывать возможность расширения базы данных. Под расширением следует понимать увеличение числа пользователей, объема хранимых данных;
- простота администрирования;
- надежность;
- наличие единого геометрического типа данных для точек, линий, полигонов и т.д, представляющего данные в системе координат для сферической Земли;
- высокое быстродействие;
- наличие подробной документации.

Так как в качестве способа организации данных была выбрана реляционная база данных, необходимо рассмотреть и сравнить современные реляционные СУБД: MySQL, Microsoft SQL Server, Oracle Database и PostgreSQL.

В таблице 3.2 приведены результаты сравнительного анализа СУБД.

Таблица 3.2 – Сравнительный анализ СУБД

Требование	MySQL	Microsoft SQL Server	Oracle Database	PostgreSQL
Масштабируемость	+	+	+	+
Простота администрирования	+	+	-	-
Надежность	-	-	+	+
Наличие единого геометрического типа данных	-	-	+	-
Высокое быстродействие	+	+	+	-
Наличие подробной документации	+	+	+	-
Итого	4	4	5	2

На основе результата проведенного анализа СУБД была выбрана СУБД Oracle. Данная СУБД позволяет работать с геометрическими данными благодаря компоненту Oracle Spatial. Хранения данных в Oracle Spatial имеет множество преимуществ:

- индустриальный формат. Oracle Spatial поддерживается в качестве стандартного формата хранения всеми крупными ГИС-вендорами и поставщиками пространственных данных;
- скорость работы. Обработка данных осуществляется в месте их хранения, что очень эффективно и масштабируемо;
- централизованное хранение, от этого выигрывает безопасность и надёжность хранения;
- преимущества самой СУБД Oracle. Oracle – самая мощная СУБД, потому как в неё встроены мощные возможности обработки данных: отказоустойчивость, шифрование, репликация, транзакционность и т.д.

3.4 Разработка физической модели данных географической информационной системы

Следующим этапом разработки СУБД является разработка диаграммы физической модели данных.

Физическая модель представляет собой логическую модель данных, выраженную при помощи терминов языка описания данных конкретной СУБД. Физическая модель базы данных содержит элементы, необходимые СУБД для создания базы: наименования таблиц и столбцов, типы данных, первичные и внешние ключи. Диаграмма физической модели данных представлена на рисунке 3.2.

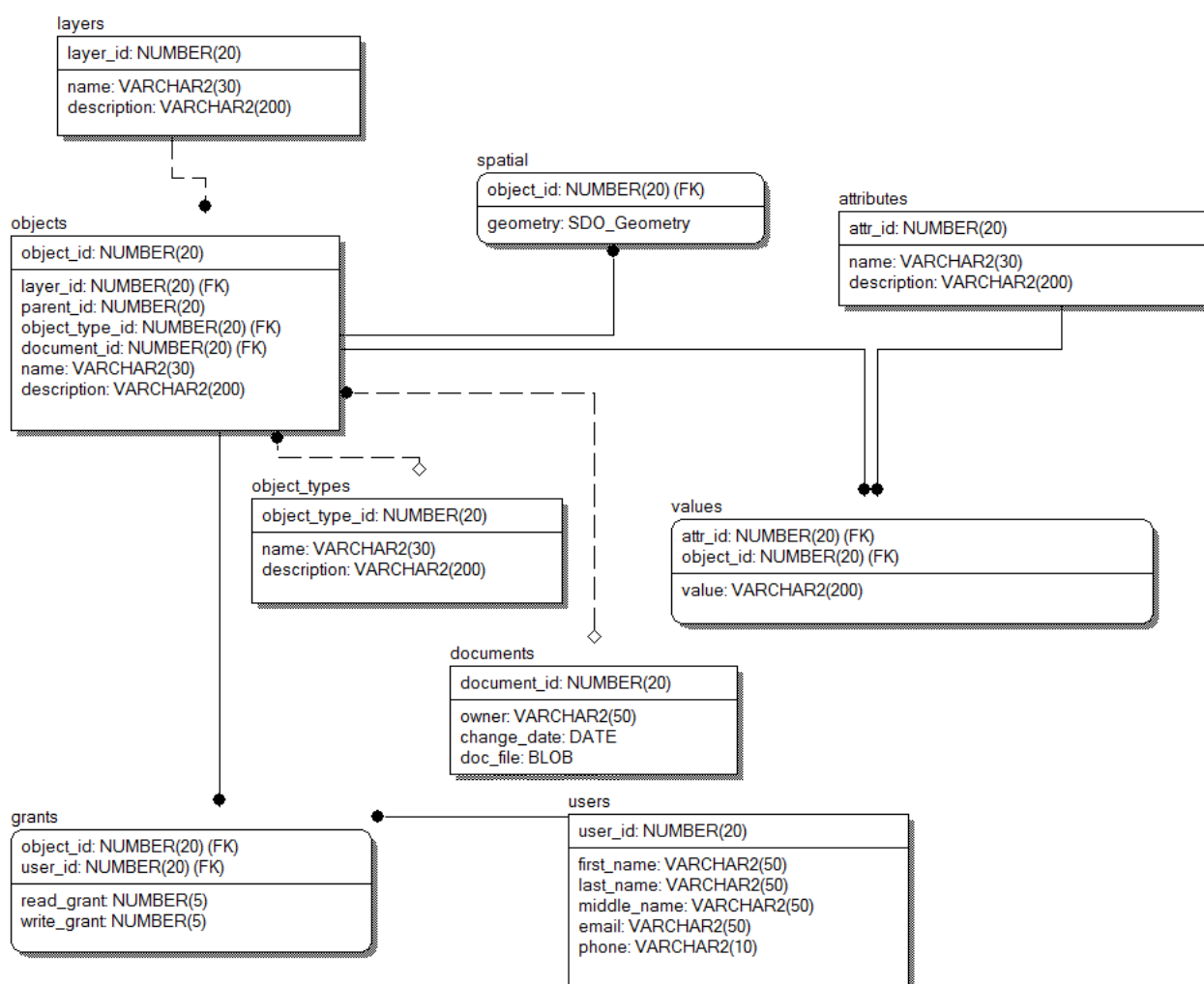


Рисунок 3.2 – Диаграмма физической модели данных

Представленная на рисунке 3.2 модель содержит в себе всю информацию, необходимую для реализации в СУБД Oracle.

3.5 Разработка программного обеспечения географической информационной системы

Разработку системы следует начать с создания REST сервисов. REST - это общие принципы организации взаимодействия приложения/сайта с сервером посредством протокола HTTP. Особенность REST в том, что сервер не запоминает состояние пользователя между запросами - в каждом запросе передаётся информация, идентифицирующая пользователя (например, token, полученный через OAuth-авторизацию) и все параметры, необходимые для выполнения операции.

Архитектура современного Spring приложения подразумевает наличие сущностей, сервис-слоя, доменного слоя и слоя представления.

Разрабатываемая система включает следующие сущности: Document, GeoObject, Message, User, Grants и другие.

Сущность Message представлена на рисунке 3.3.

```
@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Message implements Serializable {
    @Column
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Id
    private Integer messageId;
    @Column
    private String message;
    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.EAGER)
    private List<GeoJsonObject> geoJsonObjects = new ArrayList<>();
}
```

Рисунок 3.3 – Сущность Message

Для генерации конструкторов и полей используются аннотации @Data и

`@NoArgsConstructor` плагина Lombok. Lombok — это плагин компилятора, который добавляет в Java новые «ключевые слова» и превращает аннотации в Java-код, уменьшая усилия на разработку и обеспечивая некоторую дополнительную функциональность. Для обозначения того факта, что сущность является JPA сущностью, данный класс имеет аннотацию `@Entity`.

Реализация сервис-слоя подразумевает создание класса, содержащего в себе бизнес-логику работы с данными. Классы сервис-слоя помечаются аннотацией `@Service`. Класс сервис-слоя `MessageServiceImpl` отвечает за работу с данными сообщений пользователей, прикрепленных к географическому положению (рисунок 3.4).

```
@Service
public class MessageServiceImpl implements MessageService {

    private final MessageRepository messageRepository;

    @Autowired
    public MessageServiceImpl(MessageRepository messageRepository) { this.messageRepository = messageRepository; }

    @Override
    public MessageDTO getMessageById(Integer messageId) {
        return MessageConverter.entityToDto(messageRepository.getOne(messageId));
    }

    @Override
    public void saveUser(MessageDTO userDto) { messageRepository.save(MessageConverter.dtoToEntity(userDto)); }

    @Override
    public List<MessageDTO> getAllUsers() {
        return messageRepository.findAll().stream().map(MessageConverter::entityToDto).collect(Collectors.toList());
    }
}
```

Рисунок 3.4 – Класс сервис-слоя

В системе присутствуют следующие классы сервис-слоя: `GeoService`, отвечающий за работу с геоданными, `GrantService`, отвечающий за работу с правами пользователей, `DocumentService` для работы с документами. Также разработаны сервисы для работы со сторонними источниками данных.

В качестве доменного слоя в системе выступают классы `MessageRepository`, `UserRepository`, `GeoRepository` и другие. На рисунке 3.5 представлен класс `MessageRepository`.

```

@Repository
@CrossOrigin(origins = "http://localhost:4200")
public interface MessageRepository extends JpaRepository<Message, Integer> {
}

```

Рисунок 3.5 – Класс доменного слоя

Классы доменного слоя помечаются аннотацией `@Repository`. Аннотация показывает, что класс функционирует как репозиторий и требует наличия прозрачной трансляции исключений. Преимуществом трансляции исключений является то, что слой сервиса будет иметь дело с общей иерархией исключений, вне зависимости от используемых технологий доступа к данным в слое данных.

Для работы веб-сервиса необходимо создать класс, аннотированный как `@Controller`, `@ResponseBody` или `@RestController` (рисунок 3.6).

```

@RestController
@RequestMapping("/objects")
public class GeoObjectController {
    private final GeoService geoService;

    @Autowired
    public GeoObjectController(GeoService geoService) { this.geoService = geoService; }

    @RequestMapping(method = RequestMethod.GET)
    public Collection<GeoObject> getAllObjects() { return geoService.getAllObjects(); }

    @RequestMapping(value =("/{id}", method = RequestMethod.DELETE)
    public void deleteGeoObjectById(@PathVariable("id") int id) { geoService.removeGeoObjectById(id); }

    @RequestMapping(method = RequestMethod.PUT, consumes = MediaType.APPLICATION_JSON_VALUE)
    public void deleteGeoObjectById(@RequestBody GeoObject geoObject) { geoService.updateGeoObject(geoObject); }

    @RequestMapping(method = RequestMethod.POST, consumes = MediaType.APPLICATION_JSON_VALUE)
    public void insertGeoObject(@RequestBody GeoObject geoObject) { geoService.insertGeoObject(geoObject); }
}

```

Рисунок 3.6 – Класс слоя представления

Аннотация `@Controller` указывает на то, что класс относится к слою представления и является контроллером. Класс `GeoService` представляет собой компонент, похожий на обычный сервлет, но с расширенными возможностями от Spring Framework.

Также данный класс имеет аннотацию `@RequestMapping`. Данная аннотация используется для маппинга URL-адреса запроса на указанный метод или класс.

Правило одного источника является важным принципом обеспечения безопасности, реализованное веб-браузерами для предотвращения выполнения JavaScript кодом запросов к другим источникам, а не к тому, с которого он передан. Однако, это правило, несмотря на свою эффективность, ограничивает взаимодействие между сервером и клиентом.

Cross-Origin Resource Sharing (CORS) является техникой для ослабления правила одного источника, позволяя JavaScript на веб-странице обрабатывать ответ от REST API от другого источника.

Обычно междоменные взаимодействия начинаются с запроса GET, POST или HEAD к ресурсу на сервере. Запрос включает заголовок `Origin`, который указывает на происхождение клиентского кода. Сервер учитывает `Origin` запроса и принимает или отказывает в обработке запроса. Если сервер принял запрос, он ответит запрашиваемым ресурсом в заголовке `Access-Control-Allow-Origin`. Этот заголовок будет указывать клиенту с каким происхождением клиента будет разрешен доступ к ресурсу. Принимая во внимание, что `Access-Control-Allow-Origin` соответствует `Origin` запроса, браузер разрешит запрос. В противном случае, если `Access-Control-Allow-Origin` отсутствует в ответе или если его нет в `Origin` запроса, то браузер не разрешит запрос.

Для того, чтобы включить CORS, необходимо добавить аннотацию `@CrossOrigin` во все контроллеры запросов. Теперь клиент сможет взаимодействовать с сервером без каких-либо проблем.

Для обеспечения сохранности данных необходимо предусмотреть контроль доступа к хранимой в системе информации. Для решения задач такого вида существует Spring Security. Spring Security – это Java фреймворк, предоставляющий механизмы построения систем аутентификации и авторизации. Данный инструмент был использован для создания механизмов контроля доступа в приложении.

Для конфигурации Spring Security необходимо создать класс WebSecurityConfig, содержащий в себе метод configure(). Код класса представлен в приложении А. Далее необходимо рассмотреть использованные аннотации.

Аннотация EnableWebSecurity включает поддержку web security и обеспечивает интеграцию со Spring MVC. Метод configure(HttpSecurity) определяет, какой URL нужно защищать, а какой не надо (по умолчанию все защищено) Вызов методов antMatchers("/", "/base").permitAll() указывает, какие urls защищать не надо. Вызов метода loginPage("/login") определяет страницу для авторизации, которая доступна всем. Вызов методов logout().permitAll() определяет, что «разлогиниться» могут все.

Далее необходимо разработать Angular компоненты, которые будут взаимодействовать с сервером и отображать необходимые данные пользователю.

Компоненты Angular состоят из файла класса компонента, файла шаблона компонента и файла стилей компонента. Код компонента, отвечающего за отображение карты на странице представлен в приложении Б.

Файл шаблона представляет собой html файл, который заполняется данными перед отображением в браузере. Таким образом, при открытии страницы пользователем выполняется метод из файла класса компонента ngOnInit(), инициализирующий коллекцию маркеров и создающий карту. Далее коллекция маркеров отображается на странице при помощи шаблонизатора.

3.6 Описание функциональности географической информационной системы

Для запуска геоинформационной системы необходимо запустить веб-браузер (Mozilla Firefox, Google Chrome). В адресной строке ввести адрес системы. Далее при запуске Системы в окне браузера появляется главное окно программы.

При старте система позволяет производить следующие манипуляции: перемещение по карте, масштабирование, вывод карты на печать, поиск по объектам и просмотр информации в одном из пяти слоев (рисунок 3.7).

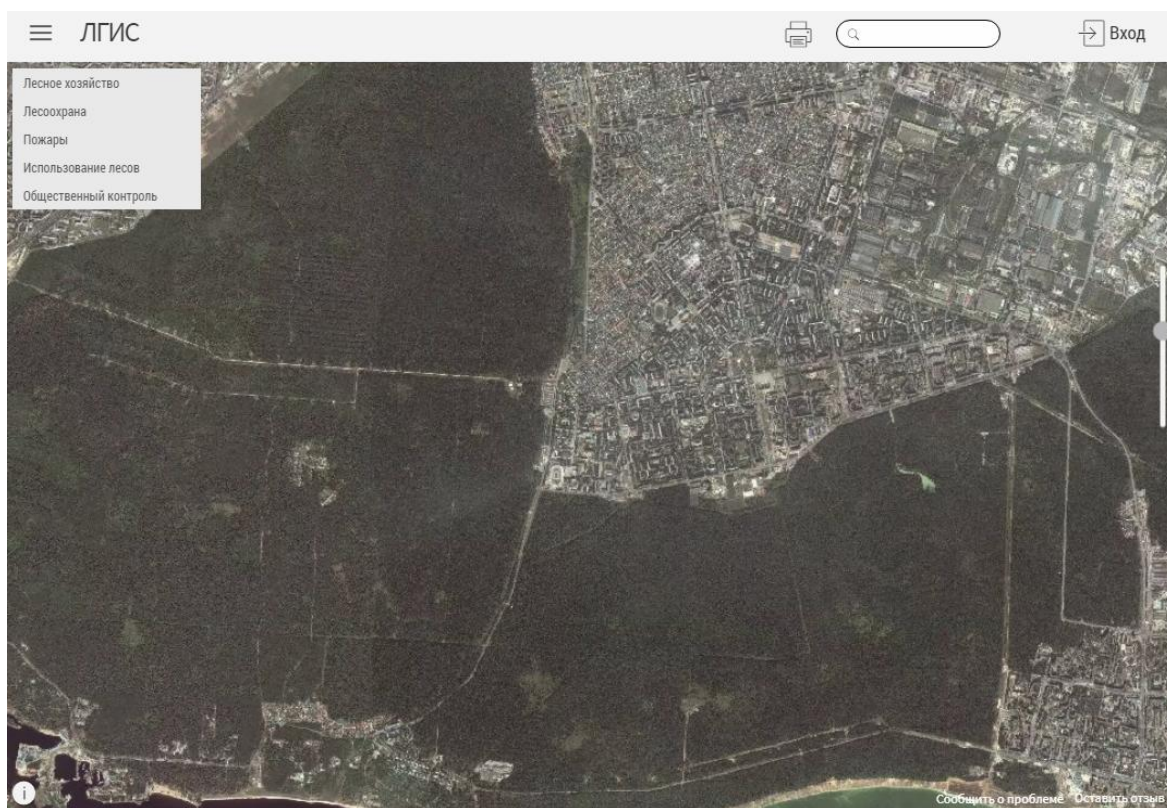


Рисунок 3.7 – Главное окно программы

Для перемещения по карте пользователю необходимо навести указатель мыши на участок карты, нажать левую клавишу, и, не отпуская ее, переместить указатель в нужном направлении. Для изменения масштаба можно использовать колесо мыши или ползунок, расположенный в окне справа.

Для упрощения поиска объектов в системе предусмотрена функция «поиск». Чтобы найти необходимый объект, необходимо ввести часть его названия или название целиком в поле, располагающееся сверху, в «шапке» страницы, рядом с кнопкой входа, и нажать кнопку Enter на клавиатуре.

Пользователь может продолжать работу без авторизации, либо ввести свои имя и пароль для того, чтобы авторизоваться (рисунок 3.8).

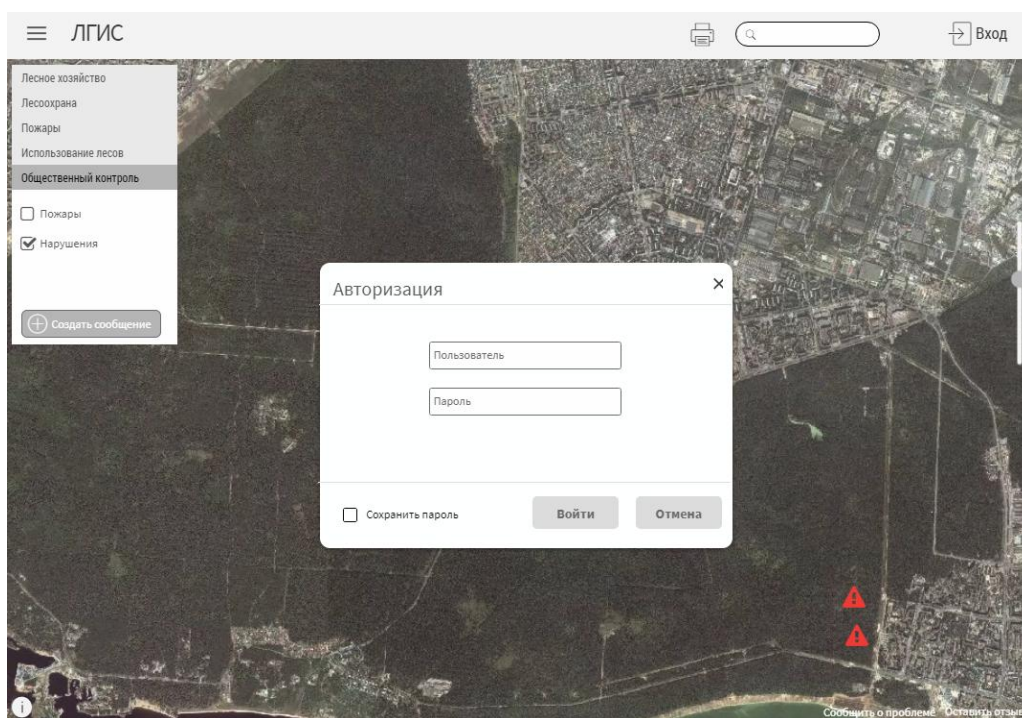


Рисунок 3.8 – Окно авторизации

В неавторизованном режиме пользователю доступны все функции системы, за исключением редактирования. Однако, пользователь может создать сообщение о нарушении, воспользовавшись слоем «Общественный контроль» (рисунок 3.9).

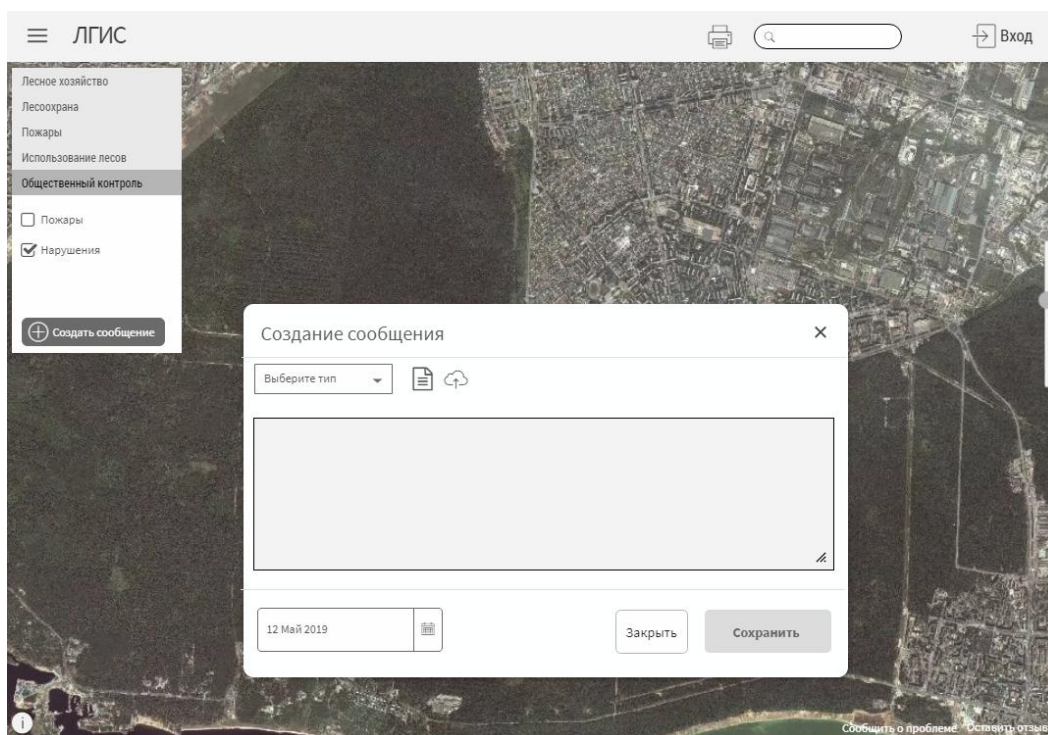


Рисунок 3.9 – Окно создания сообщения

Авторизованным пользователям доступно редактирование всех слоев данных. На рисунке 3.10 изображено окно системы с панелью инструментов, доступной авторизованному пользователю.

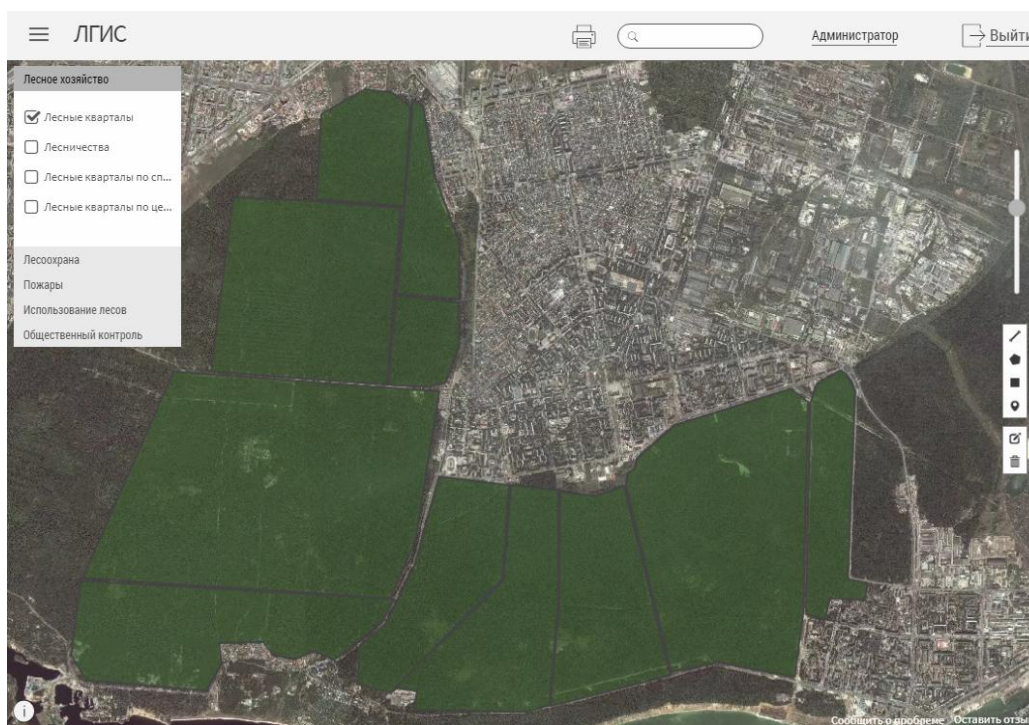


Рисунок 3.10 – Окно авторизованного пользователя

На рисунке 3.10 изображено окно системы с открытым слоем «лесное хозяйство» и отмеченным чекбоксом «лесные кварталы», изображенные в виде зеленых многоугольников.

Для того, чтобы выйти из системы, пользователь должен нажать кнопку «Выйти» в правом верхнем углу страницы.

Выводы по главе 3

В данной главе был произведен выбор архитектуры географической информационной систем, технологии разработки программного обеспечения и СУБД. После этого была построена физическая модель базы данных.

Затем была произведена разработка веб-приложения согласно поставленным ранее требованиям и выбранным технологиям.

Завершающим этапом была разработка пользовательского интерфейса и описание функционала серверной и клиентской частей приложения.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был проанализирован и описан существующий бизнес-процесс лесничества г. Тольятти. После разбиения процесса учета лесного фонда на конкретные этапы с использованием функциональной декомпозиции, стали видны конкретные этапы процесса, являющиеся наиболее трудоёмкими. Для этого была применена технология структурного анализа, на основе которого были построены диаграммы IDEF0.

Необходимость уточнения информации о лесном фонде, стихийные бедствия, сложность природных процессов роста и развития насаждений требуют постоянной корректировки данных. Однако, данный процесс требует больших трудозатрат при текущем бизнес-процессе, на основании чего было принято решение об автоматизации бизнес-процесса ведения учета лесного хозяйства путем внедрения географической информационной системы.

В дальнейшем были определены требования к системе и произведен анализ известных решений. Анализ показал, что ни одна существующая система не соответствует всем предъявленным к требованиям, на основании чего было принято решение о проектировании и разработке географической информационной системы.

Далее была проанализирована учебно-методическая и научная литература, необходимая для разработки географической информационной системы, произведено логическое и физическое моделирование географической информационной системы, проектирование базы данных.

В результате работы была спроектирована и реализована географическая информационная система, осуществляющая автоматизацию деятельности инженеров лесного хозяйства, а именно процесс учета лесного фонда.

Благодаря автоматизации процесса учета лесного фонда сократится количество ресурсов и времени, затрачиваемых сотрудниками лесничества.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Исаев, Г.Н. Проектирование информационных систем. Учебное пособие. - М.: Омега-Л, 2015. - 432с.
2. Карвин, Б. Программирование баз данных SQL. Типичные ошибки и их устранение / Б. Карвин. – М.: Рид Групп, 2012. – 336с. – (Профессиональные компьютерные книги).
3. Репин В. В., Бизнес-процессы. Моделирование, внедрение, управление. – М.: Манн, Иванов и Фербер, 2013. – 512 с.
4. Коваленко, В.В. Проектирование информационных систем. - М.: Форум, 2012. - 320с.
5. Сеттер, Р.В. Изучаем Java на примерах и задачах – Наука и техника, 2016. – 240 с.
6. Вязовик, Н. А. Программирование на Java, Интуит 2016. – 600 с.
7. Александров, Д.В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы / Д.В. Александров. - М.: Финансы и статистика, 2011. – 224 с.
8. Джошуа Блох. Java. Эффективное программирование: Лори, 2014. – 440с.
9. ER-Win Data Modeler. Режим доступа - https://ru.wikipedia.org/wiki/ERwin_Data_Modeler.
10. Обзор современных реляционных СУБД. Режим доступа - https://spravochnick.ru/bazy_dannyh/yazyk_sql_osnovy_raboty_s_relyacionnymi_sudbd_osnovy_yazyka_sql/obzor_sovremennyh_relyacionnyh_sudbd/
11. Об информации, информационных технологиях и о защите информации /Федеральный закон от 27 июля 2006 г. № 149-ФЗ // Справочная правовая система «Гарант». – 2014. – 16 янв.

12. Золотов С. Ю. Проектирование информационных систем: учеб. пособие / С. Ю. Золотов; Томский гос. ун-т систем управления и радиоэлектроники. - Томск: Эль Контент, 2013. - 86 с

Электронные ресурсы

13. Главная страница [Электронный ресурс]: Официальный сайт «Министерство природных ресурсов и экологии». – Режим доступа: <http://www.mpr.krskstate.ru/>.

14. Все о ГИС и их применении [Электронный ресурс] Геоинформационные системы и технологии – Режим доступа: <http://gistechinik.ru>.

15. Лесной кодекс Российской Федерации [Электронный ресурс]: федеральный закон от 04.12.2006 № 200 - ФЗ ред. от 01.05.2016. // Справочная правовая система «Консультант Плюс». Режим доступа: <http://www.consultant.ru>.

16. Государственный лесной реестр Российской Федерации, [Электронный ресурс] федер. закон от 04.12.2006 № 200 –ФЗ ред. от 01.05.2016. // Справочная правовая система «Консультант Плюс». Режим доступа: <http://www.consultant.ru>.

17. Как выбрать технологию для большого и не очень большого веб-проекта [Электронный ресурс]: статья. Режим доступа: <https://tproger.ru/articles/which-technology-to-choose/>.

18. Тренды в языках программирования 2019 [Электронный ресурс]: статья. Режим доступа: <https://habr.com/ru/post/431654/>.

19. Выбор технологий для большого и не очень большого веб-проекта [Электронный ресурс]: статья. Режим доступа: https://habr.com/ru/company/SECL_GROUP/blog/315734/.

Литература на иностранном языке

18. Nevner, Alan, Chatterjee, Samir. Design Research in Information Systems - Theory and Practice, 2010.

19. Dhillon, G. (2007). Principles of Information Systems Security: Text and Cases. Hoboken, NJ: John Wiley & Sons. 73

20. Nyussupova G.N. Free and open source GIS software [Электронный ресурс]: educational manual/ Nyussupova G.N., Kairova Sh.G., Kalimurzina A.M. — Электрон. текстовые данные. — Алматы: Казахский национальный университет им. аль-Фараби, 2014. — 84 с. — Режим доступа: <http://www.iprbookshop.ru/59734.html>. — ЭБС «IPRbooks»

21. ER-Win Data Modeler. Режим доступа - https://ru.wikipedia.org/wiki/ERwin_Data_Modeler.

22. Mario Fusco, Alan Mycroft. Java 8 in Action: Lambdas, Streams, and functional-style programming. – 1st Edition, Manning Publications, 2014

23. Craig Walls. Spring in Action. – Fifth Edition, Manning Publications, 2018

ПРИЛОЖЕНИЕ А

```
@Configuration
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .authorizeRequests()
            .antMatchers("/", "/base").permitAll()
            .anyRequest().authenticated()
            .and()
            .formLogin()
            .loginPage("/login")
            .permitAll()
            .and()
            .logout()
            .permitAll();
    }
}
```


ПРИЛОЖЕНИЕ Б

Компонент map-box.

Файл map-box.component.html:

```
<input type="text" [(ngModel)]="message">
<div *ngFor="let marker of markers">
  <button (click)="flyTo(marker)">{{ marker.properties.message }}</button>
  <button (click)="removeMarker(marker)">Delete</button>
</div>
<div class="map" id="map"></div>
```

Файл map-box.component.scss:

```
#map {
  height: 800px;
  width: 100%;
}
html,
body {
  font-size: 12px;
}
```

Файл map-box.component.ts:

```
import { Component, OnInit } from "@angular/core";
import * as mapboxgl from "mapbox-gl";
import { MapService } from "../map.service";
import { GeoJson, FeatureCollection } from "../map";
import { AngularFireDatabase, AngularFireList } from
"@angular/fire/database";

@Component({
```

```

selector: "app-map-box",
templateUrl: "./map-box.component.html",
styleUrls: ["./map-box.component.scss"]
})

export class MapBoxComponent implements OnInit {
  map: mapboxgl.Map;
  style = "mapbox://styles/mapbox/satellite";
  lat = 37.75;
  lng = -122.41;
  message = "Hello World!";

  source: any;
  markers: any = [];

  constructor(
    private mapService: MapService,
    private db: AngularFireDatabase
  ) {}

  ngOnInit() {
    this.markers = this.mapService
      .getMarkers()
      .valueChanges()
      .subscribe(markers => {
        this.markers = markers;
        console.log(markers);
      });
    console.log(this.markers);

    this.initializeMap();
  }
}

```

```
}
```

```
private initializeMap() {  
  if (navigator.geolocation) {  
    navigator.geolocation.getCurrentPosition(position => {  
      this.lat = position.coords.latitude;  
      this.lng = position.coords.longitude;  
      this.map.flyTo({  
        center: [this.lng, this.lat]  
      });  
    });  
  }  
  this.buildMap();  
}
```

```
buildMap() {  
  this.map = new mapboxgl.Map({  
    container: "map",  
    style: this.style,  
    zoom: 13,  
    center: [this.lng, this.lat]  
  });  
  
  this.map.addControl(new mapboxgl.NavigationControl());  
  
  this.map.on("click", event => {  
    const coordinates = [event.lngLat.lng, event.lngLat.lat];  
    const newMarker = new GeoJson(coordinates, { message: this.message });  
    this.mapService.createMarker(newMarker);  
  });  
}
```

```

this.map.on("load", event => {
  this.map.addSource("geoService", {
    type: "geojson",
    data: {
      type: "FeatureCollection",
      features: []
    }
  });

  this.source = this.map.getSource("geoService");

  this.markers.valueChanges().subscribe(markers => {
    let data = new FeatureCollection(markers);
    this.source.setData(data);
  });

  this.map.addLayer({
    id: "geoService",
    source: "geoService",
    type: "symbol",
    layout: {
      "text-field": "{message}",
      "text-size": 24,
      "text-transform": "uppercase",
      "icon-image": "rocket-15",
      "text-offset": [0, 1.5]
    },
    paint: {
      "text-color": "#f16624",

```

```
        "text-halo-color": "#fff",
        "text-halo-width": 2
    }
});
});
}

removeMarker(marker) {
    this.mapService.removeMarker(marker.$key);
}

flyTo(data: GeoJson) {
    this.map.flyTo({
        center: data.geometry.coordinates
    });
}
}
```