

Министерство науки высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(Университет ИТМО)
Факультет информационных технологий и программирования

ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

по теме:

**ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ РАСЧЕТНОГО МОДУЛЯ
ДЛЯ СИСТЕМЫ АВТОМАТИЧЕСКОГО ФОРМИРОВАНИЯ
ГЕНЕРАЛЬНЫХ ПЛАНОВ
ПЛОЩАДНЫХ ОБЪЕКТОВ КАПИТАЛЬНОГО СТРОИТЕЛЬСТВА**

СПИСОК ИСПОЛНИТЕЛЕЙ

Научный руководитель,
Университет ИТМО,
факультет информационных технологий и программирования,
преподаватель

Пантенков С. А.

Студент,
Направление подготовки 09.04.02
Информационные системы и технологии,
Академическая группа М42051

Степанов С. В.

Санкт-Петербург

2022

Реферат

Данный отчет состоит из 16 страниц. Содержит в себе 3 иллюстрации, 11 использованных источников. В данном отчете проводится анализ и обоснование применяемых технических решений в рамках реализации системы автоматического формирования генеральных планов площадных объектов капитального строительства. Проводится анализ имеющихся ресурсов для реализации системы, обосновывается выбор технологического стека и выбор компонентов архитектуры системы.

Содержание

Определения, обозначения и сокращения	4
Введение	5
Постановка задачи	6
Актуальность проблемы	7
Требования к системе	7
Функциональные требования	8
Нефункциональные требования	8
Технические решения	9
Выбор стека технологий	9
Выбор архитектуры системы	11
Заключение	15
Список литературы	16

Определения, обозначения и сокращения

В настоящем отчете о НИР применяют следующие термины с соответствующими определениями.

Генеральный план (генплан, ГП) в общем смысле — проектный документ, на основании которого осуществляется планировка, застройка, реконструкция и иные виды градостроительного освоения территорий. Основной частью генерального плана (также называемой собственно генеральным планом) является масштабное изображение, полученное методом графического наложения чертежа проектируемого объекта на топографический, инженерно-топографический или фотографический план территории.

Площадными объектами капитального строительства (ПО) в данной работе называются здания, строения, сооружения, а также объекты, строительство которых не завершено, за исключением некапитальных строений, сооружений и неотделимых улучшений земельного участка (замощение, покрытие и другие)[1].

Здание — результат строительства, представляющий собой объемную строительную систему, имеющую надземную и (или) подземную части, включающую в себя помещения, сети инженерно-технического обеспечения и системы инженерно-технического обеспечения и предназначенную для проживания и (или) деятельности людей, размещения производства, хранения продукции или содержания животных[2].

Сооружение — результат строительства, представляющий собой объемную, плоскостную или линейную строительную систему, имеющую наземную, надземную и (или) подземную части, состоящую из несущих, а в отдельных случаях и ограждающих строительных конструкций и предназначенную для выполнения производственных процессов различного вида, хранения продукции, временного пребывания людей, перемещения людей и грузов[2].

Строения — общее понятие зданий и сооружений[3].

Введение

В современном мире каждая компания хочет минимизировать свои затраты, как по времени, так и в финансовом плане. Это не обошло и сферу строительства. В данной области существует ряд открытых задач, требующих решения, в том числе это касается сферы проектирования генеральных планов площадных объектов.

Любое сооружение несет в себе определенную функцию. Склады служат для хранения вещей, факелы для сжигания попутного газа, а нефтеперерабатывающий завод для преобразования сырой нефти в нефтепродукты. Чем сложнее задача, которую решает промышленный объект, тем сложнее он устроен, и из большого числа сооружений он состоит. А чем больше сооружений находится рядом, тем больше появляется различных ограничений, продиктованных вопросами безопасности строительства этих зданий рядом.

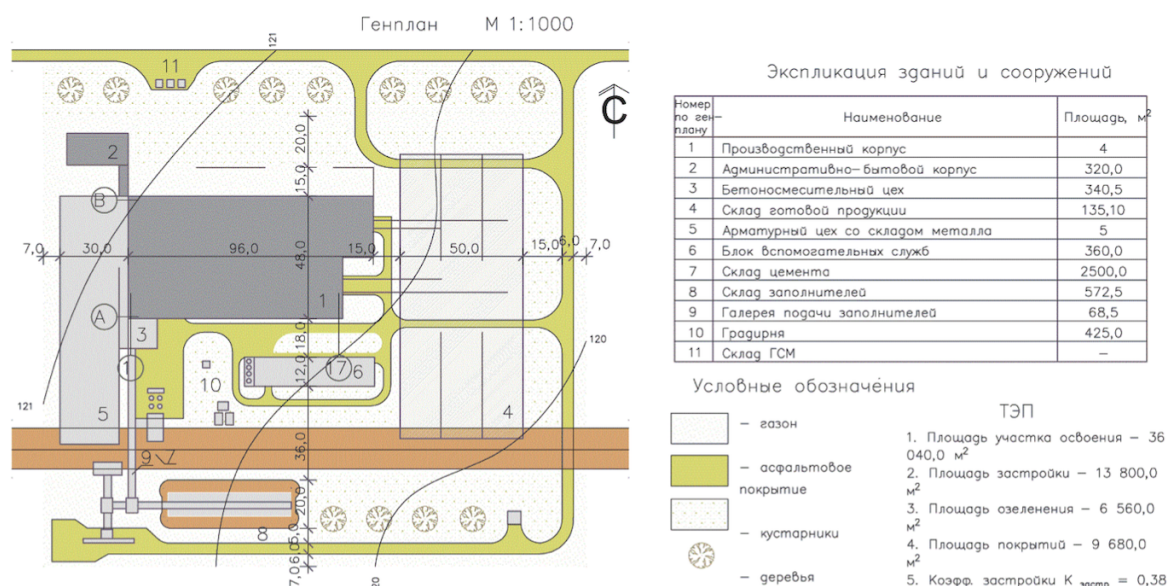


Рис. 1. Генеральный план завода

Крайне опасно размещать факел для сжигания попутного рядом с хранилищем топлива из-за высокой пожароопасности хранилища. В случае возгорания хранилища, топливо, которое находится в нём, может быть не только огнеопасным, но также и взрывоопасным. Соответственно, если в зоне поражения взрыва находятся сооружения, требующие постоянного присутствия людей, то помимо повреждения материальных объектов, могут погибнуть или быть травмированы люди, чего допускать никак нельзя.

Проектирование генерального плана промышленного сооружения служит, как раз для того, чтобы минимизировать риски возникновения нештатных ситуаций, повреждения материального имущества, а также, что наиболее важно, риск возникновения несчастных случаев на предприятии.

Также очень важно учитывать не только взаимодействие зданий внутри промышленного объекта, но и учитывать характеристики местности, где этот объект будет по-

строен. Характеристики местности очень разнообразны и включают в себя не только физические величины, такие как объем грунта при разравнивании площадки и общую длину ограждения вокруг объекта, так и различные юридические аспекты, будь то уровень загрязнения окружающей среды или уровень производимого шума.

Постановка задачи

Задача автоматической генерации генерального плана сооружения выглядит следующим образом. Пользователь задает входные данные для нашей системы, запускает расчет и на выходе получает планировку площадного объекта.

Список входных данных:

- допустимая для строительства область на карте;
- стоимостная модель расчета стоимости инженерной подготовки;
- перечень сооружений с указанием габаритов (ширина, длина, радиус), степени огнестойкости, категории взрывопожарной и пожарной опасности, конструктивной пожарной опасности;
- параметры коммуникаций между сооружениями проектируемого объекта;
- местоположения внешних точек подключения площадного объекта (дорога, ВЛ, трубопроводы внешнего транспорта и т.п.);
- параметры цифровой модели рельефа (ЦМР);

В качестве результата работы системы получаем:

- фигуру площадного объекта, содержащей заданные сооружения;
- местоположения сооружений;
- схемы технологических эстакад минимальной длины, соединяющей все сооружения, введенные пользователем;
- схему внутриплощадочных проездов;
- стоимость инженерной подготовки, коэффициента застройки территории и длины технологических связей;
- зоны распространения теплового потока;
- зоны распространения взрывной волны;

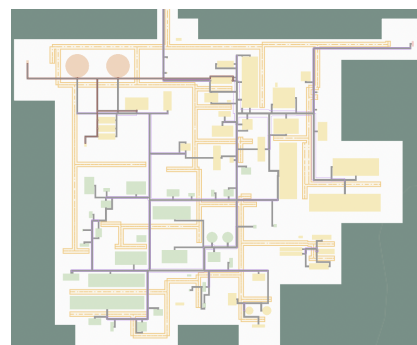


Рис. 2. Пример результата работы

Система используется, как площадка для исследований и демонстрации различных алгоритмических подходов для решения задачи, так как полноценного решения математической задачи еще не существует. Это ведет к тому, что существующая методика решения задачи может в одночасье стать неактуальной, потому что найдется новая методика, которая будет давать более качественное решение. Поэтому система должна быть очень гибкой, чтобы без больших затрат со стороны разработки вносить подобные изменения.

Целью данной работы является анализ и обоснование применяемых технических решений в рамках реализации данной системы.

Исходя из этой цели можно выделить следующие *задачи*:

1. проанализировать имеющиеся ресурсы для реализации системы,
2. выбрать и обосновать технологический стек,
3. спроектировать архитектуру системы в целом и обосновать выбор компонентов системы,
4. обосновать способ взаимодействия компонентов системы друг с другом.

Актуальность проблемы

Основной задачей системы является получения генерального плана площадного объекта в автоматическом режиме на основе заданного перечня сооружений, технологических и инженерных связей, формы и размера участка местности, доступного для застройки, а также требований нормативной документации. Создание инструмента для аналитики и формирования генеральных планов площадных объектов позволит снизить капитальные затраты при строительстве промышленных объектов благодаря повышению качества проектирования. Подобный инструмент позволит сравнить несколько различных вариантов планировок одного промышленного объекта, чтобы впоследствии выбрать оптимальный по ряду критериев, а также получить обоснование, почему выбранный вариант планировки действительно является оптимальным.

Данная задача является чрезвычайно сложной в алгоритмическом плане и не имеет готовых методик решения. Для её решения требуется провести ряд научных исследований в области алгоритмов. Процесс исследований напрямую связан с взаимодействием с техническими экспертами в области проектирования генеральных планов.

Требования к системе

В данном пункте кратко перечислим те функциональные и нефункциональные требования, с которыми пришлось столкнуться в рамках реализации расчетного модуля. Так

как объем отчета ограничен, то в целях экономии места, приведем их в менее формальном виде.

Функциональные требования

У нас есть 3 группы пользователей, которые будут взаимодействовать с системой. И каждая из этих групп пользователей имеет свои интересы.

1. Технические эксперты, представители заказчика, руководитель проекта

- хотят иметь интерактивный доступ к результатам исследований на различных кейсах,
- хотят иметь возможность сравнить результаты работы различных методик решения задачи,
- хотят иметь возможность интерактивно запустить алгоритмы на своих данных и получить результат.

2. Аналитики

- хотят оперативно видеть результаты работы команды исследователей,
- хотят проводить анализ результатов различных методик, полученных на разных этапах развития проекта.

3. Исследователи

- хотят иметь легкий доступ к данным, полученным от пользователей,
- хотят иметь простой способ для отправки результатов новой методики для дальнейшего анализа команде аналитиков,
- хотят иметь возможность оперативно добавлять новые методики в действующий функционал системы.

Нефункциональные требования

Из нефункциональных требований отметим наиболее важные, которые определяют сам процесс проектирования системы и выбора технологий для реализации.

1. *Вычислительные ресурсы.* Для заказчика очень важна сохранность данных, которые он предоставляет, поэтому одним из требований является использование собственных серверов компании, а не использование облачных решений, популярных в текущий момент в индустрии.

2. *Уровень анализа предметной области.* Главной сложностью исследовательских проектов является невозможность проработки предметной области на достаточно глубоком уровне, чтобы спроектировать систему в детальном виде, в связи с короткими сроками и отсутствием качественной экспертизы у заказчика. Необходимо предусмотреть высокий уровень гибкости системы, так как требования к ней могут меняться очень часто на противоположные.
3. *Требования к отчетной документации.* Самой главной ценностью в проекте являются алгоритмы. Именно их и требует заказчик, как основной результат нашей деятельности. Алгоритмы должны представлять собой отдельную библиотеку, которая должна быть версионизируема и быть готова передана заказчику по первому требованию.
4. *Временные ресурсы.* Основной задачей проекта является показать, что применяемый набор математических методов перспективен в плане развития. Эту перспективность требуется показать в достаточно сжатые сроки. Помимо команды исследователей, это затрагивает и команду разработки, так как требуется обеспечить качественную визуализацию результатов исследований для оперативного уточнения требований.

Технические решения

Выбор стека технологий

Во время проектирования *системы в целом* всегда возникает ряд вопросов, связанных с выбором стека технологий для решения следующих задач.

1. Реализация программного кода
2. Хранение данных
3. Развертывание
4. Тестирование
5. Логирование
6. Документация
7. Протоколы взаимодействия

Рассмотрим каждую из задач более подробно и выберем технологию, которая наиболее подходит для её решения в нашем случае.

1. Язык программирования.

В качестве основного языка программирования был выбран *Python 3.8*[4]. Самым ценным ресурсом при реализации нашей системы является время разработки. Python является языком программирования с динамической типизацией, имеет простой синтаксис, а также очень большую базу готовых библиотек для решения любых задач. Все эти преимущества позволяют, как быстро реализовывать функционал со стороны разработки, так и проводить проверку разных алгоритмических гипотез за короткое время.

Такие недостатки Python, как низкая скорость выполнения программ и большое потребление памяти, в нашем случае не играют решающей роли[5]. В Ubuntu 20.04 LTS – операционной системе, на которой ведется разработка, установлен по умолчанию Python 3.8, поэтому была выбрана именно эта версия языка.

2. Хранение данных.

Все данные, которые используются в проекте, достаточно однородны и могут быть легко представлены в табличном виде. Поэтому в качестве технологии для хранения данных будем использовать реляционную базу данных. Так как проектирование генеральных планов напрямую связано с геоданными, то необходимо учитывать это требование.

В качестве технического решения предлагается связка *PostgreSQL 12* и расширения для работы с геоданными *PostGIS 3.1*. PostgreSQL является популярным open-source решением в мире баз данных, а PostGIS содержит очень большое количество функций для преобразований геоданных, что нам очень важно[6]. В версии PostGIS 3.0 появилась поддержка Mapbox Vector Tile, что позволяет генерировать данные для визуализации с помощью стандартных функций базы данных.

3. Развертывание.

Необходимо предусмотреть механизм автоматического развертывания. В компании используется система управления git-репозиториями *GitLab*. В этой системе доступен набор инструментов CI/CD, позволяющих решить данные задачи. Он называется *GitLab-CI*. Так как разработка ведется на языке Python, то самым удобным способом развертывания приложений является технология Docker[8].

А для объединения нескольких *Docker*-контейнеров вместе можно воспользоваться средствами *docker-compose*. Более сложные инструменты оркестрации контейнеров, такие как *kubernetes* нам не требуются, в силу того, что у нас только один вычислительный сервер.

4. Тестирование.

Наша система состоит из нескольких компонентов. Поэтому нужно предусмотреть следующие виды тестирования[7].

- *Модульное тестирование.* Для проверки корректности работы различных функций и методов.
- *Функциональное тестирование.* Для проверки соответствия результатов методов бизнес-

требованиям.

- *Интеграционное тестирование.* Для проверки, что все компоненты системы работают согласованно.

5. Логирование.

Помимо логирования в коде, требуется подумать о системе сбора, агрегации и хранения логов. Одним из лучших решений на рынке является ELK. Это решение состоит из трех компонент: Elasticsearch для полнотекстового поиска и аналитики, Logstash для сбора логов из приложений, Kibana для визуализации логов в веб-интерфейсе.

6. Документация.

Документацию имеет смысл писать только тогда, когда она действительно будет использоваться. У нас таких случаев ровно два: документация для внешнего API и документация для математической библиотеки.

В первом случае, есть множество стандартных средств таких, как Swagger или GraphiQL, которые позволяют генерировать документацию на основе модели данных автоматически.

Во втором случае требуется сгенерировать описание алгоритмов, используемых в программе. Важная особенность в документировании алгоритмов заключается в том, что так как они зачастую очень сложны в понимании только по текстовому описанию и требуют наличия иллюстраций. Это удобно делать в Markdown-файлах. Также нужно учесть, что в алгоритмах присутствует не только теоретическое обоснование, но и реализация, которая также требует документации, но уже стандартными средствами разработки.

Чтобы объединить в себе два формата документации, теоретическую часть и реализацию, воспользуемся инструментом Sphinx[9].

7. Протоколы взаимодействия.

Самым удобным средством визуализации данных для пользователя является веб-интерфейс. Его главное преимущество заключается в том, что веб-интерфейс доступен на любом устройстве и не требует установки дополнительных программ, достаточно лишь веб-браузера для просмотра. В качестве формата взаимодействия между веб-интерфейсом и сервером выберем формат JSON, так как это один из наиболее популярных стандартов в индустрии. Также он является человеко-читаемым и легко редактируемым с помощью обычного текстового редактора, в отличие от protobuf.

Выбор архитектуры системы

Для решения задачи автоматической генерации генерального плана сооружения предлагается следующая архитектура системы.

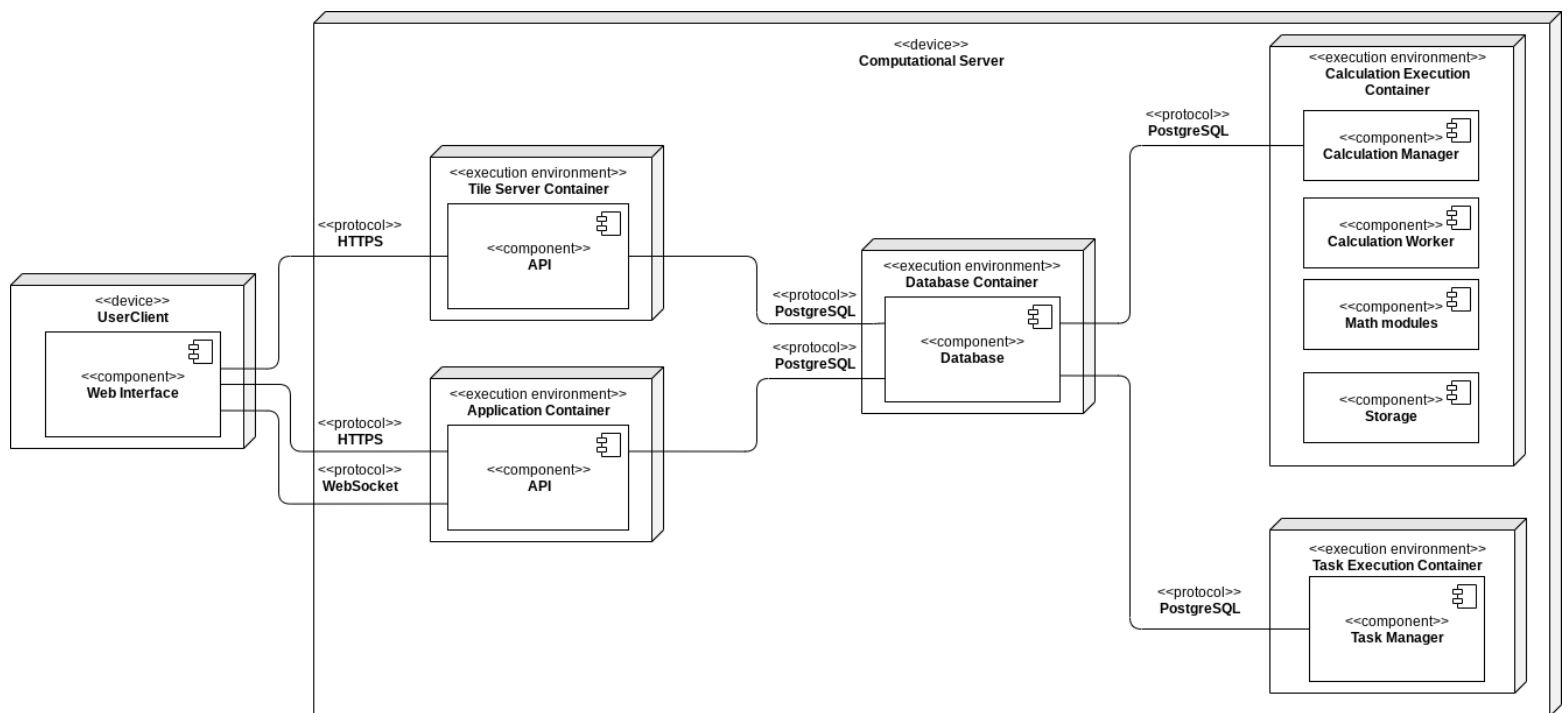


Рис. 3. Диаграмма размещения

Компоненты представленные на диаграмме:

- **Web Interface** – веб-интерфейс, позволяющий осуществлять взаимодействие с системой пользователю.
- **Tile Server Container**
 - API – веб-сервер, предоставляющий API для получения подложки карты, а также векторные данные для повышения производительности отображения данных на карте.
- **Application Container**
 - API – веб-сервер, предоставляющий GraphQL-API для взаимодействия с системой. Также отвечает за предоставление и обработку данных, обеспечивающих выполнение расчета.
- **Database** – база данных (БД), используемая для хранения данных расчетных задач.
- **Task Execution Container**
 - Task Manager – компонент, контролирующий расчет задачи. Последовательно вызывает этапы выполнения задачи.
- **Calculation Execution Container**
 - Calculation Manager – компонент, осуществляющий получение и преобразование данных из БД к данным, обрабатываемых Calculation Worker-ами, а также за их запуск.

- Calculation Worker – компонент, осуществляющий запуск метода в отдельном процессе, обработку и сохранение решения.
- Math Modules – модули, представляющие собой математическую библиотеку, которая используется Calculation Worker-ом.
- Storage – компонент, осуществляющий чтение/запись входных данных для метода.

Расчет генплана площадного объекта – это комплексная задача, которая состоит из множества различных, связанных между собой этапов. Результатом задачи является построенный генеральный план. Генеральный план состоит из следующих объектов:

- Местоположение сооружений.
- Коммуникационные сети (линии электропередач, трубопроводы).
- Эстакады для прокладки коммуникационных сетей.
- Внутриплощадочные проезды.
- Местоположение площадного объекта на местности.
- Зоны теплового излучения.
- Зоны поражения взрывных волн.
- Объёмы выемки/отсыпки строительной площадки.
- Ограждения групп сооружений.

Вычисление всех этих объектов представляет последовательный вызов расчетных методов. Каждый из этих методов выполняет свою определенную задачу: один рассчитывает местоположения зданий, другой, на основе местоположений зданий рассчитывает конфигурацию внутриплощадочных проездов. Каждый из методов влияет на результаты других методов. Поэтому разная последовательность их вызовов может давать кардинально разные результаты.

Расчет каждого метода может продолжаться длительное время (например: больше одного часа). Поэтому очень важно чтобы результаты расчетов методов для задачи были сохранены. В случае возникновения ошибки в процессе расчета задачи, нельзя допустить, чтобы результаты успешно пройденных этапов были потеряны. Требуется, чтобы после исправления ошибки в расчете можно было продолжить выполнение задачи.

Поэтому расчет задачи выполняется в двух отдельных контейнерах: *Task Execution Container* отвечает за оркестрацию вызываемых методов для расчета, а *Calculation Execution Container* за запуск методов из математической библиотеки в отдельных процессах.

Для того чтобы снизить временные издержки при разработке, будем упрощать интерфейсы взаимодействия между компонентами. Так как все данные хранятся в базе данных, то нам потребуется реализовать интерфейс взаимодействия между БД и программным кодом. Данный интерфейс можно будет переиспользовать в трех контейнерах, что позволит сэкономить время на разработку.

Так как каждый контейнер – это отдельный процесс, то при запуске задач мы решаем проблему межпроцессного взаимодействия. Самый простой интерфейс сетевого взаимодействия между двумя процессами – это использование таблицы в базе данных. Мы можем применить данный способ потому что у нас малое количество рассчитываемых задач, поэтому проблемы с использованием БД, как брокера сообщений, просто не возникает.

Заключение

В данной работе был проведен анализ и сделан выбор технических решений для реализации системы решения задачи автоматической генерации генерального плана площадного объекта. Было составлено обоснование выбора технологического стека для решения данной задачи. В рамках выбранного технологического стека и с учетом требований была спроектирована архитектура системы, а так же выполнено обоснование проекта данной архитектуры и применения компонент, которые в ней используются.

Список литературы

1. «Градостроительный кодекс Российской Федерации» от 29.12.2004 N 190-ФЗ (ред. от 31.12.2017)
2. Федеральный закон «Технический регламент о безопасности зданий и сооружений» от 30.12.2009 N 384-ФЗ.
3. «Гражданский кодекс Российской Федерации (часть первая)» от 30.11.1994 N 51-ФЗ (ред. от 06.04.2011), Статья 233
4. Python 3.8 Documentation [Интернет ресурс]:
URL:<https://docs.python.org/3.8/> (дата обращения: 28.01.22)
5. Python Advantages and Disadvantages [Интернет ресурс]:
URL:<https://techvidvan.com/tutorials/python-advantages-and-disadvantages/>
(дата обращения: 28.01.22)
6. PostGIS Reference [Интернет ресурс]:
URL:<https://postgis.net/documentation/> (дата обращения: 28.01.22)
7. Glenford J. Myers, Corey Sandler, Tom Badgett "The Art of Software Testing 3rd Edition, 16 Dec 2011
8. Docker Reference [Интернет ресурс]:
URL:<https://docs.docker.com> (дата обращения: 28.01.22)
9. Sphinx Python Documentation [Интернет ресурс]:
URL:<https://www.sphinx-doc.org/en/master/> (дата обращения: 28.01.22)
10. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides "Design Patterns: Elements of Reusable Object-Oriented Software 1st Edition
11. Martin Robert C. "Clean Code: A Handbook of Agile Software Craftsmanship" Aug 1, 2008