МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет  
имени академика С.П. Королева»  
(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики

Кафедра программных систем

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**«ТЕМА ПОЛНОСТЬЮ, КАК В ПРИКАЗЕ»**

по направлению подготовки 02.03.02

Фундаментальная информатика и информационные технологии

(уровень бакалавриата)

направленность (профиль) «Информационные технологии»

Студент И.И. Иванов

Руководитель ВКР

к.т.н., доцент П.П. Петров

Консультант (при наличии) С.С. Сидоров

Нормоконтролёр Е.В. Сопченко

Самара 2018МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет  
имени академика С.П. Королева»  
(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики

Кафедра программных систем

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Н. Коварцев

\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

# ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ (бакалавр)

студенту (ке) Иванов Иван Иванович

группа 6413-020302D

Тема работы: полное название, как в приказе

Цель работы (и/или исходные данные)

Структурные части работы (перечень вопросов, подлежащих разработке):



|  |  |
| --- | --- |
| Научный руководитель доцент, кафедра программных систем | Задание принял к исполнению |
| (П.П. Петров)  « » 2017 г. | (И.И. Иванов)  « » 2017 г. |

ПРИЛОЖЕНИЕ

к заданию на выпускную квалификационную работу бакалавра

студенту(ке) И.И. Иванову группа № 6413-020302D

Тема: «Полное название темы как в приказе»

Исходные данные к работе:

1. Характеристики объекта автоматизации:
2. объект автоматизации – ;
3. виды автоматизируемой деятельности:

* процесс ;
* процесс;
* процесс;

1. минимальное количество параметров ???? – 2;
2. максимальное количество параметров ??? – 3;
3. минимальное количество параметров для многомерных методов – 2;
4. количество способов отображения функции для одномерной оптимизации– 2;
5. количество ролей пользователей – 3;
6. количество выполняемых работ (занятий) – 2;
7. количество студентов в группе – не более 30;
8. количество групп – не менее 2;
9. количество преподавателей – не менее 2;
10. количество вариантов заданий – не более 100.
11. Требования к информационному обеспечению:
12. Информационное обеспечение разрабатывается на основании следующих литературных источников:

* Банди Б. Методы оптимизации. Вводный курс: Пер. с англ. М.: Радио и связь, 1988. 128 с.
* Мокрова Н.В., Суркова Л.Е. Методы оптимизации: учебное пособие. М: МГУИЭ, 2006. 90 с.
* Лемешко Б.Ю. Методы оптимизации: конспект лекций. Новосибирск: Изд-во НГТУ, 2009. 126 с.
* Мицель А.А., Шелестов А.А. Методы оптимизации: учебное пособие. Томск: изд-во Томского межвузовского центра дистанционного образования, 2002. 192 с.

и следующих сведений:

* сведений о студентах;
* сведений о преподавателях.

1. Результаты выполнения заданий должны сохраняться в файлы формата \*.doc.
2. Обеспечить контроль целостности базы данных.
3. Предусмотреть резервное копирование системы и данных на машинный носитель.
4. Предусмотреть защиту информации от несанкционированного доступа.
5. В системе должны вестись следующие справочники:

* справочник учебных групп;
* справочник видов функций;
* справочник тестовых функций.

1. Требования к техническому обеспечению:
2. Требования к серверной части:

* тип ЭВМ – IBM PC совместимый;
* тактовая частота процессора – не менее 1.8 ГГц;
* объем оперативной памяти – не менее 512 МБ;
* объём свободного дискового пространства – не менее 3 ГБ.

1. Требования к клиентской части:

* тип ЭВМ – IBM PC совместимый;
* монитор с разрешающей способностью не ниже 1280 х 786;
* тактовая частота процессора – не менее 1 ГГц;
* объем оперативной памяти – не менее 256 МБ;
* протокол обмена данными ‑ TCP/IP;
* объём свободного дискового пространства – не менее 2 ГБ;
* клавиатура;
* манипулятор – мышь.

1. Требования к программному обеспечению:
2. Требования к серверной части:

* операционная система – Windows Server 2008 R2;
* СУБД – MySQL 5.1;
* .Net Framework 4.0;
* среда программирования – Microsoft Visual Studio 2012;
* виртуальная среда – MySQL Workbench;
* язык программирования – C# 4.0.

1. Требования к клиентской части:

* операционная система – MS Windows XP и выше.

1. Общие требования к проектируемой системе.

5.1 Функции, реализуемые системой:

1. Функции администратора:

* ;
* ;
* определение прав доступа;
* ;

1. Функции преподавателя:

* ;
* ;
* ;
* ;
* ;

1. Функции студента:

* ;
* ;
* ;
* ;

1. Общесистемные функции:

* авторизация и аутентификация пользователей в системе;
* визуализация результатов работы;
* выдача справочной информации о системе.

5.2 Технические требования к системе:

1. режим работы – диалоговый;
2. температура окружающего воздуха – 15-25°С;
3. влажность окружающего воздуха – 45-75%;
4. система должна удовлетворять санитарным правилам и нормам СанПин 2.2.2/2.4.2198-07;
5. условия работы средств вычислительной техники должны соответствовать ГОСТ 12.1.005, 12.1.007.

Научный руководитель,   
к.т.н, доцент П.П. Петров

РЕФЕРАТ

Пояснительная записка: ХХ с, ХХ рисунка(ов), ХХ таблиц(ы), ХХ источников, ХХ приложение(я).

Графическая часть: ХХ слайдов презентации PowerPoint.

КЛЮЧЕВЫЕ СЛОВА, КЛЮЧЕВЫЕ СЛОВА, КЛЮЧЕВЫЕ СЛОВА, КЛЮЧЕВЫЕ СЛОВА, КЛЮЧЕВЫЕ СЛОВА, КЛЮЧЕВЫЕ СЛОВА.

Объектом исследования является …

Цель работы – …

В процессе работы была разработана автоматизированная система…, позволяющая … Система разработана на языке программирования ХХХ с использованием ХХХ (перечислить фрейморки, библиотеки и т.п.) и функционирует под управлением операционной системы ХХХ. Доступ к данным осуществляется с помощью СУБД ХХХ.

СОДЕРЖАНИЕ

[Введение 10](#_Toc507082777)

[1 Описание и анализ предметной области 11](#_Toc507082778)

[1.1 Основные понятия и определения 11](#_Toc507082779)

[1.2 Описание систем-аналогов 12](#_Toc507082780)

[1.2.1 Пункт (может не иметь заголовок) 12](#_Toc507082781)

[1.2.2 Пункт (может не иметь заголовок) 13](#_Toc507082782)

[1.3 Постановка задачи 13](#_Toc507082783)

[2 Проектирование системы 14](#_Toc507082784)

[2.1 Структурная схема системы 14](#_Toc507082785)

[2.2 Разработка прототипа интерфейса пользователя 14](#_Toc507082786)

[2.3 Информационно-логический проект системы 14](#_Toc507082787)

[2.3.1 Диаграмма вариантов использования 15](#_Toc507082788)

[2.3.2 Диаграмма состояний 15](#_Toc507082789)

[2.3.3 Диаграмма деятельности 15](#_Toc507082790)

[2.3.4 Диаграмма последовательности 15](#_Toc507082791)

[2.3.5 Диаграмма классов 16](#_Toc507082792)

[2.3.6 Логическая модель данных (при необходимости) 16](#_Toc507082793)

[2.4 Выбор и обоснование комплекса программных средств 16](#_Toc507082794)

[2.4.1 Выбор и обоснование операционной системы 16](#_Toc507082795)

[2.4.2 Выбор и обоснование языка программирования 16](#_Toc507082796)

[2.4.3 Выбор и обоснование среды разработки 17](#_Toc507082797)

[2.4.4 Выбор и обоснование СУБД (при необходимости) 17](#_Toc507082798)

[2.4.5 Выбор и обоснование библиотеки ХХХ (при необходимости) 17](#_Toc507082799)

[3 Реализация системы 18](#_Toc507082800)

[3.1 Описание интерфейса пользователя 18](#_Toc507082801)

[3.2 Диаграммы реализации 18](#_Toc507082802)

[3.2.1 Диаграмма классов 18](#_Toc507082803)

[3.2.2 Диаграмма компонентов 19](#_Toc507082804)

[3.2.3 Диаграмма развертывания 19](#_Toc507082805)

[3.2.4 Физическая модель данных (при необходимости) 19](#_Toc507082806)

[3.3 Выбор и обоснование комплекса технических средств 19](#_Toc507082807)

[3.3.1 Расчет объема занимаемой памяти 19](#_Toc507082808)

[3.3.2 Минимальные требования, предъявляемые к системе 20](#_Toc507082809)

[Заключение 21](#_Toc507082810)

[Список использованной литературы 22](#_Toc507082811)

[Приложение А Руководство пользователя 24](#_Toc507082812)

[Приложение Б Листинг программы 25](#_Toc507082813)

Введение

Создание перспективных образцов ракетно-космической и авиационной техники требует решения целого комплекса научно-технических проблем, в том числе разработки высокопроизводительных, энергоэффективных, малогабаритных и надежных сенсорных сетей при решении задач многопараметрического контроля, измерения и управления. С точки зрения устойчивости к дестабилизирующим факторам, габаритов, энергопотребления, простоты конструкции и эксплуатационной надежности наиболее перспективными для создания сенсорных сетей являются оптические, оптомеханические и волоконно-оптические датчики (ВОД) [1].

В истории волоконно-оптических датчиков трудно зафиксировать какой-либо начальный момент в отличие от истории волоконно-оптических линий связи. Первые публикации о проектах и экспериментах с измерительной техникой, в которой использовалось бы оптическое волокно, начали появляться с 1973 г [2].

К 1978 г. число исследований и разработок в Японии и других странах стало уже ощутимым. Однако в публикациях 1970-х годов термин «волоконно-оптический датчик» еще не был общепринятым. В японской технической литературе этого периода чаще всего использовался термин «измеритель на основе оптических волокон», а в статьях на английском языке «оптический датчик на волокне». Лишь в 1981 г. термин «волоконно-оптический датчик» признан всеми и окончательно утвердился после состоявшейся в 1982 г. в Лондоне первой международной конференции по волоконно-оптическим датчикам [2].

При разработке программного обеспечения будет использоваться технология Rapid Application Development (RAD) – быстрая разработка приложений. RAD предполагает небольшую команду и короткий, но тщательно проработанный производственный график. Разработка осуществляется по спирали, когда по мере создания программы уточняются и реализуются все более и более мелкие детали.

Во время проектирования будут использоваться две основные методологии: методология ООАП (Object-Oriented Analysis/Design) технология разработки программных систем, в основу которой положена объектно-ориентированная методология представления предметной области в виде объектов, являющихся экземплярами соответствующих классов [3] и методология UML [графического](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B7%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) описания для [объектного моделирования](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5&action=edit&redlink=1) в области [разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), [моделирования бизнес-процессов](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5&action=edit&redlink=1), [системного проектирования](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и отображения [организационных структур](https://ru.wikipedia.org/wiki/%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0) [4].

Цель работы – текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

1. Описание и анализ предметной области
   1. Основные понятия и определения
      1. Волоконно-оптические датчики

Волоконно-оптический датчик или сенсор (ВОС) — небольшое по размерам устройство, в котором оптическое волокно используется как в качестве линии передачи данных, так и в качестве чувствительного элемента, способного детектировать изменения различных величин [5].

Первые попытки создания сенсоров на основе оптических волокон можно отнести к середине 70-х годов ХХ в. Публикации о более или менее приемлемых разработках и экспериментальных образцах подобных сенсоров появились во второй половине 70-х годов. Однако считается, что этот тип сенсоров сформировался как одно из направлений техники только в начале 80-х годов. Тогда же появился и термин «волоконно-оптические сенсоры» (optical-fiber sensors) [6].

Волоконно-оптические сенсоры могут быть трех типов:

* точечные;
* распределенные;
* квазираспределенные.

Точечные сенсоры позволяют проводить измерения и контролировать параметры в определенной точке. Обычно сенсоры данного типа обладают довольно высокой точностью и небольшими размерами. В зависимости от типа сенсорного элемента локализация сенсоров может достигать 0,1 см2, как в случае чувствительного элемента на основе решетки Брэгга [6].

Решетка Брэгга ‒ элемент, расположенный внутри оптоволокна и имеющий большое количество точек отражения, расположенных на определенном интервале друг от друга. При прохождении через волоконную брэгговскую решетку узкого спектра света, его отражение регистрируется измерительными приборами. Если на оптоволокно воздействуют извне, деформируя его, длина волны отраженного света изменяется, что позволяет отследить изменения измеряемого параметра.

Распределенные сенсоры осуществляют постоянный контроль параметров по длине (объему) объекта в любой его точке, где установлен сенсорный световод. Принцип действия таких сенсорных систем основан на анализе изменения параметров по длине световода и на нелинейных эффектах. Недостатком распределения измеряемого параметра по длине является относительно невысокая точность измерения.

Квазираспределенный сенсор представляет собой массив точечных сенсоров, которые основаны на волоконных решетках Брэгга и объединены одним общим световодом. Каждый элемент имеет свои уникальные характеристики, что дает возможность провести анализ его параметров независимо от других элементов. Точность таких систем определяется точностью отдельных сенсоров, а массив может объединять до 100 и более элементов. Сенсорные массивы позволяют проводить мониторинг сложных объектов, инженерных сооружений, мостов, туннелей, корпусов кораблей и летательных аппаратов. Именно квазираспределенные системы по сравнению с электрическими аналогами обладают малой массой и размерами, что особенно важно для авиации и космонавтики [6].

* + 1. Линейное резервирование

Высокий уровень надежности современных сетей оптической связи обеспечивается реализацией комплекса различных мер, среди которых ключевыми являются средства полного или частичного восстановления связи в аварийных ситуациях. Традиционно для этого применялось резервирование — целенаправленное введение в систему определенной избыточности с целью увеличения степени связности отдельных ее узлов, то есть количества независимых путей передачи информации [7].

Увеличение количества передающих трактов, на которые осуществляется переключение при аварийных ситуациях, называется линейным резервированием.

В простейшем случае резервные волноводы находятся в том же кабеле, что и основные. Но это не гарантирует надежность системы. Для уменьшения риска одновременного обрыва основного и резервного кабеля, их прокладывают по разным маршрутам.

Линейное резервирование может быть осуществлено по схеме 1+1 и 1:1. В первом случае информация передается по двум трактам сразу и приемник выбирает сигнал, содержащий наименьшее количество шумов либо наиболее сильный сигнал. В схеме 1:1 передача переключается на резервный тракт сразу при возникновении неисправности в основном. Данная схема линейного резервирования представлена на рисунке 1.

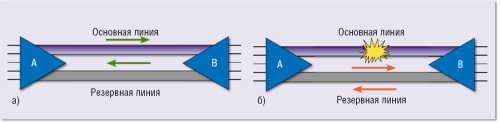
[](https://www.osp.ru/FileStorage/ARTICLE/ZHurnal_setevyh_reshenij_Telekom/2011-12/12_11/13116716/ZHurnal_setevyh_reshenij_Telekom_1_(4115).png)

Рисунок 1 ‒ Схема работы участка сети с линейным резервированием

в а) нормальном состоянии, б) при обрыве

* + 1. Кольцевые структуры

Волоконно-оптические системы часто построены на основе кольцевых топологий. В большинстве случаев линейная часть кольцевой структуры в сетях связи общего пользования строится на основе пары волокон (так называемое сдвоенное кольцо). Таким образом, сигнал может передаваться в направлении по часовой стрелке или против. Один из маршрутов исполняет роль основного тракта, а другой резервного.

Резервирование в кольцевых структурах может исполняться по схемам 1+1 и 1:1, что говорит об отсутствии отличий от линейного резервирования. В схеме 1:1 при возникновении обрыва, на узлах, находящихся на границах вышедшего из строя участка, возникает закольцовывание трафика. Пример данной схемы резервирования представлен на Рисунке 2 [7].

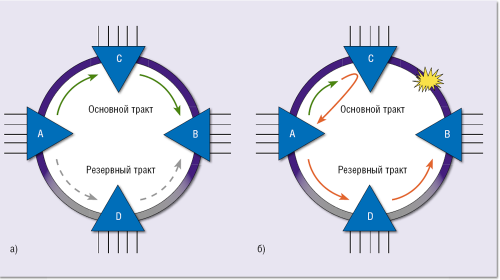
[](https://www.osp.ru/FileStorage/ARTICLE/ZHurnal_setevyh_reshenij_Telekom/2011-12/12_11/13116716/ZHurnal_setevyh_reshenij_Telekom_2_(7969).png)

Рисунок 2 ‒ Схема работы участка цепи с кольцевым резервированием в нормальном состоянии а) и после обрыва б)

* + 1. Системное резервирование

Системное резервирование представляет собой одновременное увеличение количества дополнительных волокон и приемопередающих устройств. При отказе в одном из приемопередающих устройств или при обрыве основного волокна происходит переключение на резервные волокна. Пример системного резервирования представлен на Рисунке 3.

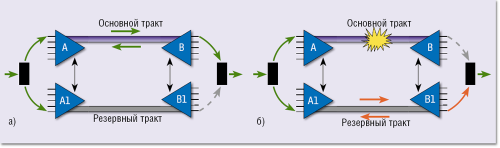
[](https://www.osp.ru/FileStorage/ARTICLE/ZHurnal_setevyh_reshenij_Telekom/2011-12/12_11/13116716/ZHurnal_setevyh_reshenij_Telekom_4_(2948).png)

Рисунок 3 ‒ Схема участка цепи с системным резервированием

* + 1. Надежность оптоволоконной сети

Надежность – свойство сети связи выполнять заданные функции, то есть обеспечивать возможность передачи требуемой информации на заданных направлениях с установленной нормами достоверностью в течение требуемого промежутка времени.

Рассмотрим оценку надежности по такому комплексному показателю, как коэффициент готовности (Кг), который характеризует безотказность и ремонтопригодность. Если оценивается надежность единичной волоконно-оптической линии связи (ВОЛС), то в случае обрыва оптического кабеля (аварии) время восстановления связи (tв) будет равняться времени восстановления данного кабеля. Если конкретный оптический кабель является фрагментом сети ВОЛС, то вполне возможно, tв будет равняться времени, необходимому для организации обходных путей. Это время в большинстве случаев значительно меньше, чем время восстановления оптического кабеля. Основными составляющими tв можно считать время: на определение места повреждения, на восстановление кабельной канализации, на прокладку аварийной вставки и на монтаж аварийной вставки с комплексом измерений. Время восстановления кабельной канализации может варьироваться от нуля до нескольких часов в зависимости от характера повреждений. Время прокладки аварийной вставки зависит от ее величины.

Коэффициент готовности (Кг) – вероятность того, что сеть связи в произвольно выбранный момент будет исправна. Кг = Т/(Т+ tв) Т – среднее время между отказами (наработка на отказ); tв – среднее время восстановления связи [8].

Четыре фактора, влияющих на коэффициент готовности:

* отказоустойчивость оборудования;
* автоматическое защитное переключение;
* методика и технологическая дисциплина эксплуатации;
* характер трассы и защитные мероприятия [9].
  + 1. Достоверность информации

Достоверность передачи данных характеризует вероятность получить искажение для передаваемого бита данных. Часто этот показатель называют интенсивностью битовых ошибок (Bit Error Rate, BER). Величина BER для каналов связи без дополнительных средств защиты от ошибок составляет, 10−4 — 10−6, в оптоволокне — 10−9. Значение BER в 10−4 говорит о том, что в среднем из 10000 бит искажается значение одного бита. Искажения бит происходят как из-за наличия помех на линии, так и по причине искажений формы сигнала ограниченной полосой пропускания линии. Для повышения достоверности передаваемых данных нужно повышать степень помехозащищённости линии, снижать уровень перекрёстных наводок в кабеле, а также использовать более широкополосные линии связи [10].

* 1. Описание систем-аналогов

Рассматриваются три системы-аналоги автоматизированных систем моделирования волоконно-оптических датчиков: «Автоматизированная система моделирования и анализа волоконно-оптических датчиков на брэгговских решетках», «Trace Mode».

* + 1. Система «Автоматизированная система моделирования и анализа волоконно-оптических датчиков на брэгговских решетках»

Интерфейс системы представлен на рисунке 4. Данная система не реализует основные функции, необходимые для моделирования датчиков с резервированием.

Можно выделить основные достоинства данной системы:

удобное отображение графиков;

большое количество параметров, позволяющее гибко настраивать объект моделирования.

Недостатками системы являются:

ограниченный функционал;

отсутствие резервирования датчиков;

отсутствие ограничения массы датчиков;

отсутствие показателя достоверности информации.

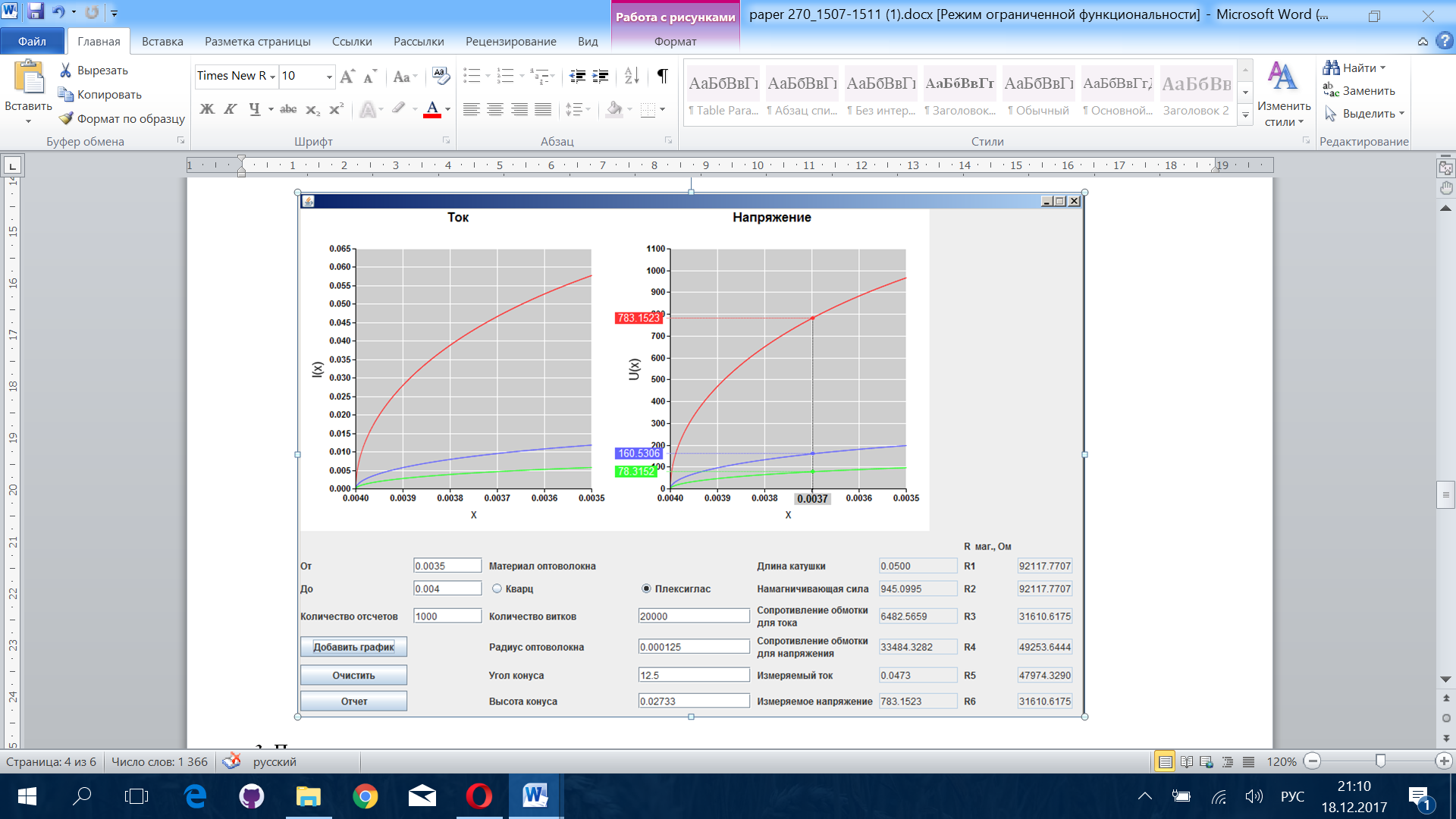


Рисунок 4 ‒ Интерфейс автоматизированной системы

* + 1. Система «Trace Mode»

Trace Mode – это высокотехнологическая программная система для автоматизации технологических процессов (АСУ ТП), телемеханики, диспетчеризации, учета ресурсов (АСКУЭ, АСКУГ) и автоматизации зданий [11]. Пример интерфейса и функционала программы представлен на рисунках 5 и 6. Данная система обладает огромным функционалом и обладает следующими достоинствами:

* большой выбор вида объекта моделирования;
* наличие функция резервирования датчиков;
* наличие развитых средств обеспечения надежности и отказоустойчивости.

Недостатками данной системы являются:

* система предоставляется платно;
* большой порог вхождения из-за наличия огромного количества функционала.

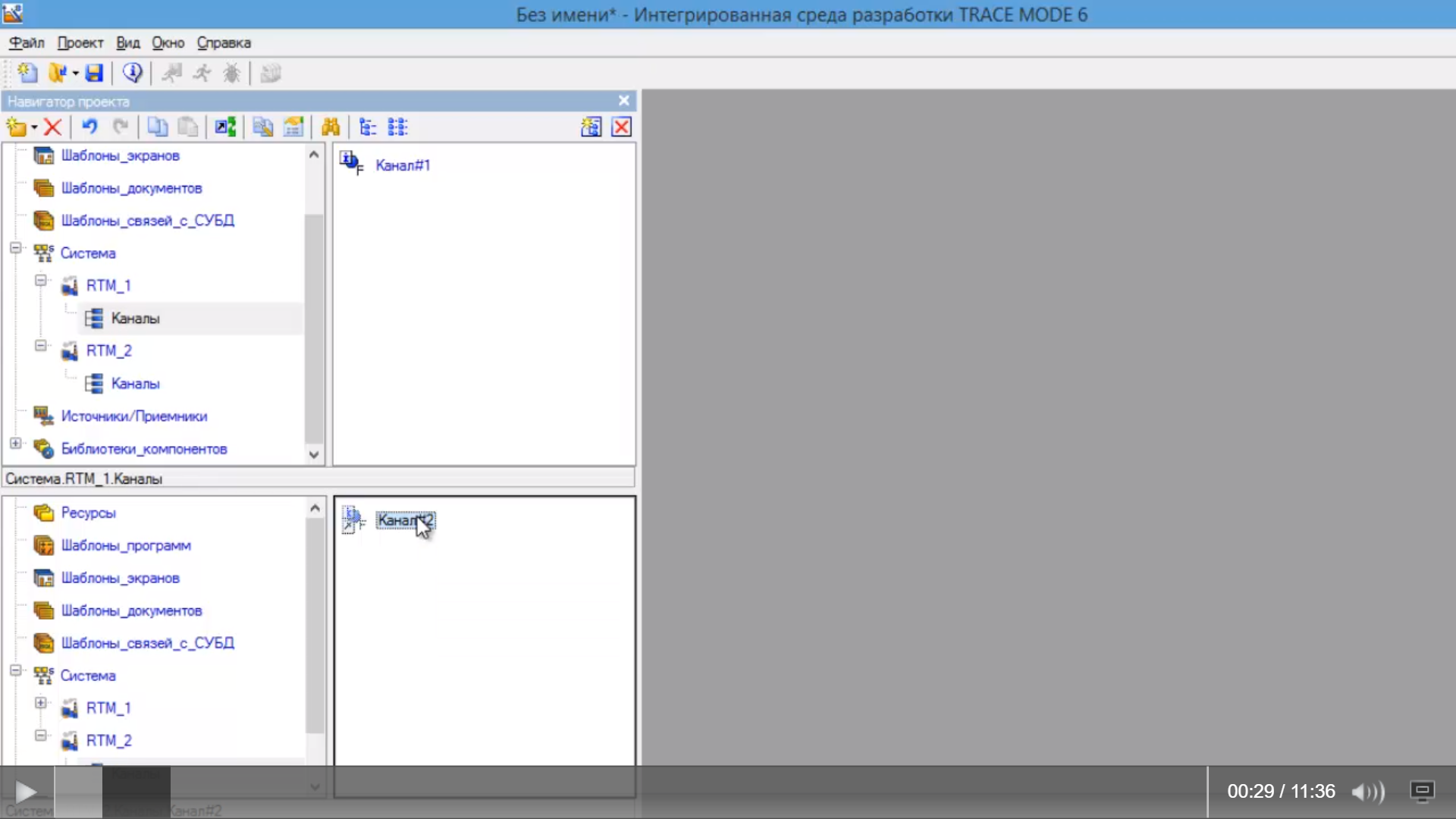


Рисунок 5 ‒ Пример интерфейса системы Trace Mode

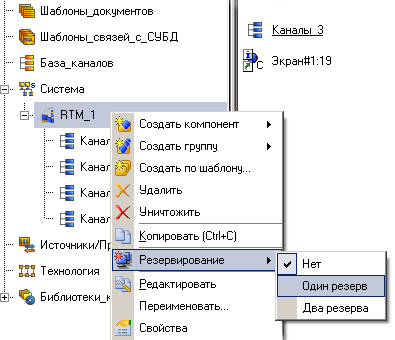


Рисунок 6 ‒ Пример фунционала системы Trace Mode

Рассмотренные системы не реализуют весь необходимый функционал и не пригодны для выполнения поставленной задачи.

* 1. Постановка задачи

Перед автором поставлена задача – разработать автоматизированную систему моделирования волоконно-оптических датчиков со структурным резервированием, в которой должны быть реализованы функции администрирования системы, управления входными и выходными параметрами, автоматического формирования параметров, моделирования волоконно-оптических датчиков и сенсорных групп с резервированием (ДСГР).

Система должна представлять собой WEB-приложение.

В системе должно быть реализовано две роли пользователей: администратор и пользователь.

*Администратор* должен авторизоваться в системе. Система должна проверить введенные данные и настроить интерфейс на данную роль. Администратор должен иметь возможность настраивать постоянные и переменные параметров моделирования ДСГР: задавать формат записи данных и параметров (количество форматов от ‒ от 2 до 6), выбирать вид отображения (количество видов ‒ от 1 до 3). Так же администратор должен добавлять и удалять входные и выходные параметры (количество входных параметров ‒ от 1 до 100, количество выходных параметров ‒ от 1 до 100). В процессе работы администратор должен сохранять модели ДСГР в файл и загружать модель ДСГР из файла (количество форматов файла ‒ от 1 до 6).

*Пользователю* не нужно регистрироваться в системе. Ему будет доступна возможность моделировать ДСГР, предварительно выбрав учитывающиеся варианты резервирования (количество вариантов резервирования ‒ от 1 до 100). Так же пользователь должен иметь возможность реализовать алгоритм введения изменений в модель ДСГР в выбранном варианте резервирования, проводить вычислительный эксперимент с математической моделью ДСГР, оценивать текущие изменения достоверности информации и надежности при изменении параметров резервирования, проводить сравнительный анализ полученных результатов с результатами натурного эксперимента.

1. *Общесистемные функции:*
   * авторизация пользователя в системе (ввод логина/пароля);
   * аутентификация пользователя в системе;
   * визуализация процессов формирования параметров надежности и достоверности;
   * автоматическое формирование параметров надежности и достоверности по заданным входным параметрам;
   * проверка корректности вводимых параметров;
   * выдача справочной информации о системе;
2. *функции администратора:*
   * настройка постоянных и переменных параметров при моделировании ДСГР:
3. задание формата записи данных и параметров;
4. выбор вида отображения.
   * редактирование входных и выходных параметров:
5. добавление параметра;
6. удаление параметра;
   * сохранение модели ДСГР в файл заданной структуры;
   * загрузка модели ДСГР из файла;
7. *функции пользователя:*

– моделирование ДСГР с учетом имеющихся и накапливаемых в базе данных вариантов резервирования, основанных на использовании нового физико-математического подхода в определения параметров надежности и оценки достоверности данных;

– реализация алгоритма введения изменений в модель ДСГР в выбранном варианте резервирования;

– проведение вычислительного эксперимента с математической моделью ДСГР;

– оценка текущего изменения достоверности информации при изменении параметров резервирования;

– оценка текущего изменения надежности ДСГР при изменении параметров резервирования;

– отображение в графическом виде процесса моделирования и результатов анализа;

– возможность сравнительного анализа полученных результатов с результатами натурного эксперимента.

1. Проектирование системы
   1. Структурная схема системы

Структурная схема представляет собой совокупность элементарных звеньев объекта и связей между ними. Схема предназначена для отражения общей структуры программы, то есть её основных блоков, частей и главных связей между ними. Из структурной схемы должно быть понятно, зачем нужно данное приложение и что оно делает в основных режимах работы, как взаимодействуют его части [10].

Приведем структурную схему разрабатываемой системы и описание некоторых подсистем (рисунок ).

Система состоит из следующих подсистем:

1. Подсистема моделирования, отвечающая за непосредственное проведения моделирования, в которую входят подсистема вычислений, выполняющая расчет степени резервирования, и подсистема сбора данных, которая собирает данные после вычислений для отчета.
2. Подсистема работы с топологиями, отвечающая добавление, изменение и удаление топологий, в которую входит подсистема конструирования топологий, которая отвечает за ввод и размещение Брэгговских решеток и оптоволокна.
3. Подсистема управления БД, отвечающая за связь с базой данных, в которой хранится информация о топологиях.
4. Подсистема визуализации, отвечающая за отображение интерфейса пользователя и предоставление доступа к элементам управления системой.
5. Подсистема ввода/вывода, отвечающая непосредственно за предоставление пользователю возможности вводить данные и видеть ответные данные системы.
6. Справочная подсистема, отвечающая за предоставление пользователю справки о работе с системой.

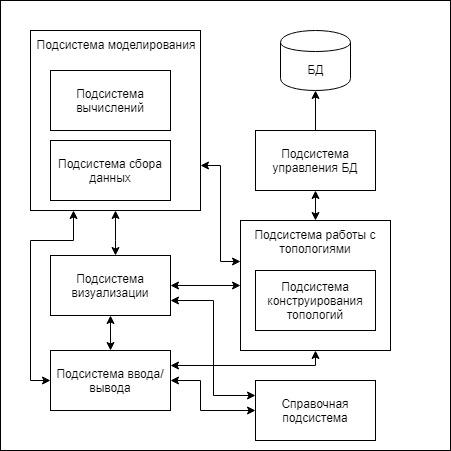


Рисунок ‒ Структурная схема

* 1. Разработка прототипа интерфейса пользователя

Интерфейс пользователя является одним из важнейших элементов программы, это та часть программы, которая находится у всех на виду. Недочеты в пользовательском интерфейсе могут серьезно испортить впечатление даже о самых многофункциональных программах. Именно поэтому разработке и проектированию пользовательского интерфейса нужно уделять особое внимание.

Некоторые программисты склонны оставлять дизайн интерфейса пользователя на потом, считая, что реальное достоинство приложения ‑ его программный код, который и требует большего внимания. Однако часто возникает недовольство пользователей из-за неудачно подобранных шрифтов, непонятного содержимого экрана и скорости его прорисовывания, поэтому работу над интерфейсом также нужно воспринимать серьезно.

Процесс создания интерфейса начинается с определения целей проекта, а также внутренних и внешние обстоятельств, которые вы должны принять во внимание. Для того чтобы правильно расставить приоритеты, необходимо учитывать:

* опыт работы пользователей с компьютером, типовые ситуации использования;
* какая информация необходима и когда, какие результаты должны быть получены;
* технологию разработки и платформа, на которой будут работать пользователи [9].

Для выполнения начальной фазы разработки необходимо погрузиться целиком в задачи пользователей и создать прототип навигационной модели. Навигационная модель показывает, как необходимо распределять функции или задачи между окнами приложения, она определяет, как пользователи смогут перемещаться как между различными задачами, так и внутри отдельной задачи [9]. Навигационная модель показывает, как необходимо распределять функции или задачи между окнами программы, она определяет, как пользователи смогут перемещаться между различными задачами и внутри отдельной задачи. На рисунке приведена навигационная модель для разрабатываемой системы.

На рисунке представлен прототип главной страницы, которая должна содержать соответствующее программе изображение, название и кнопки «Моделирование» и «Топологии». Пользователь может перейти в соответствующий раздел, нажав одну из этих кнопок. На рисунке представлен прототип страницы работы с топологиями. Она должна содержать выпадающий список топологий, каждую из которых можно выбрать, панель с отображением выбранной топологии, кнопки «Изменить»,

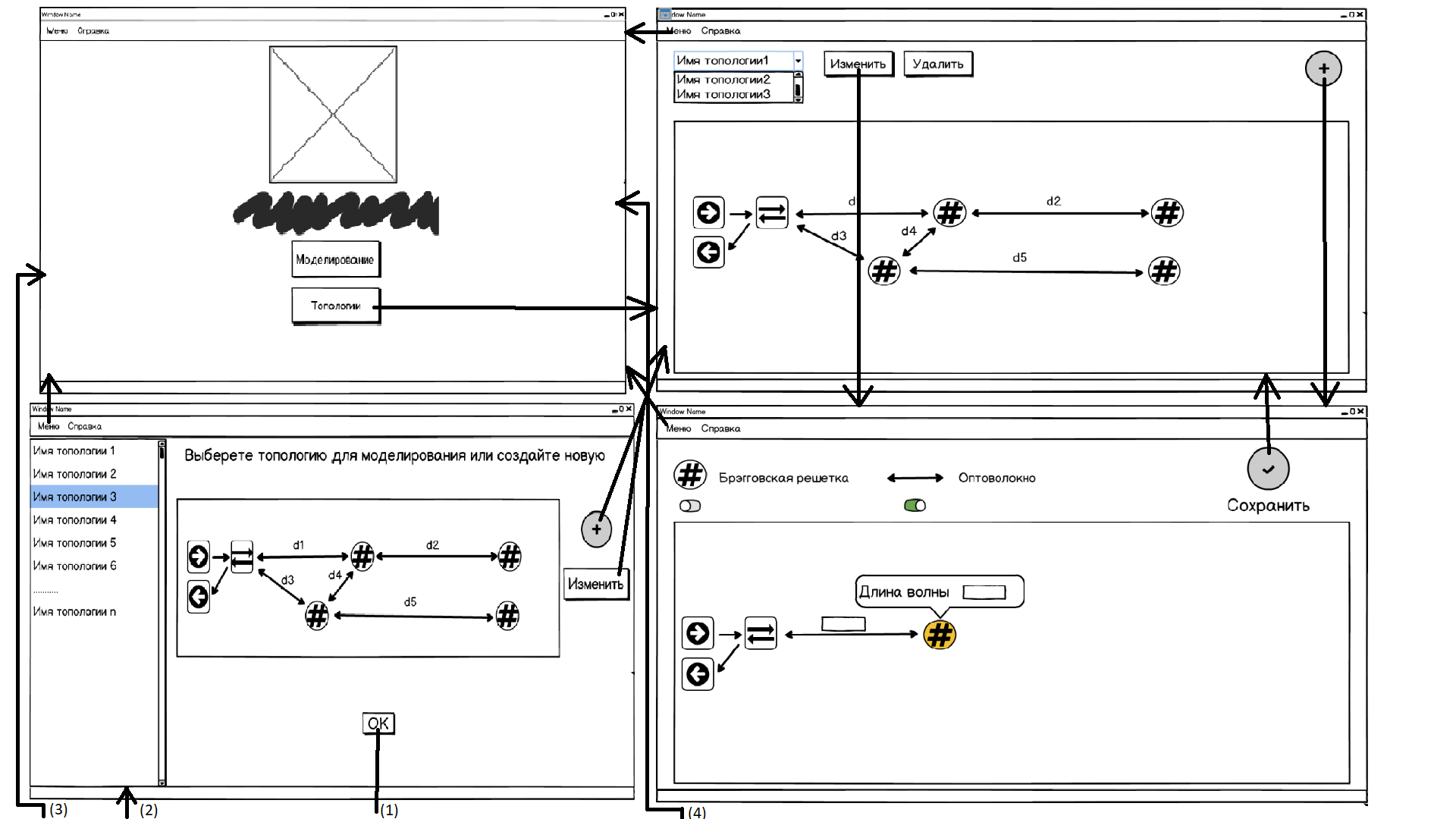


Рисунок ‒ Навигационная модель (начало)

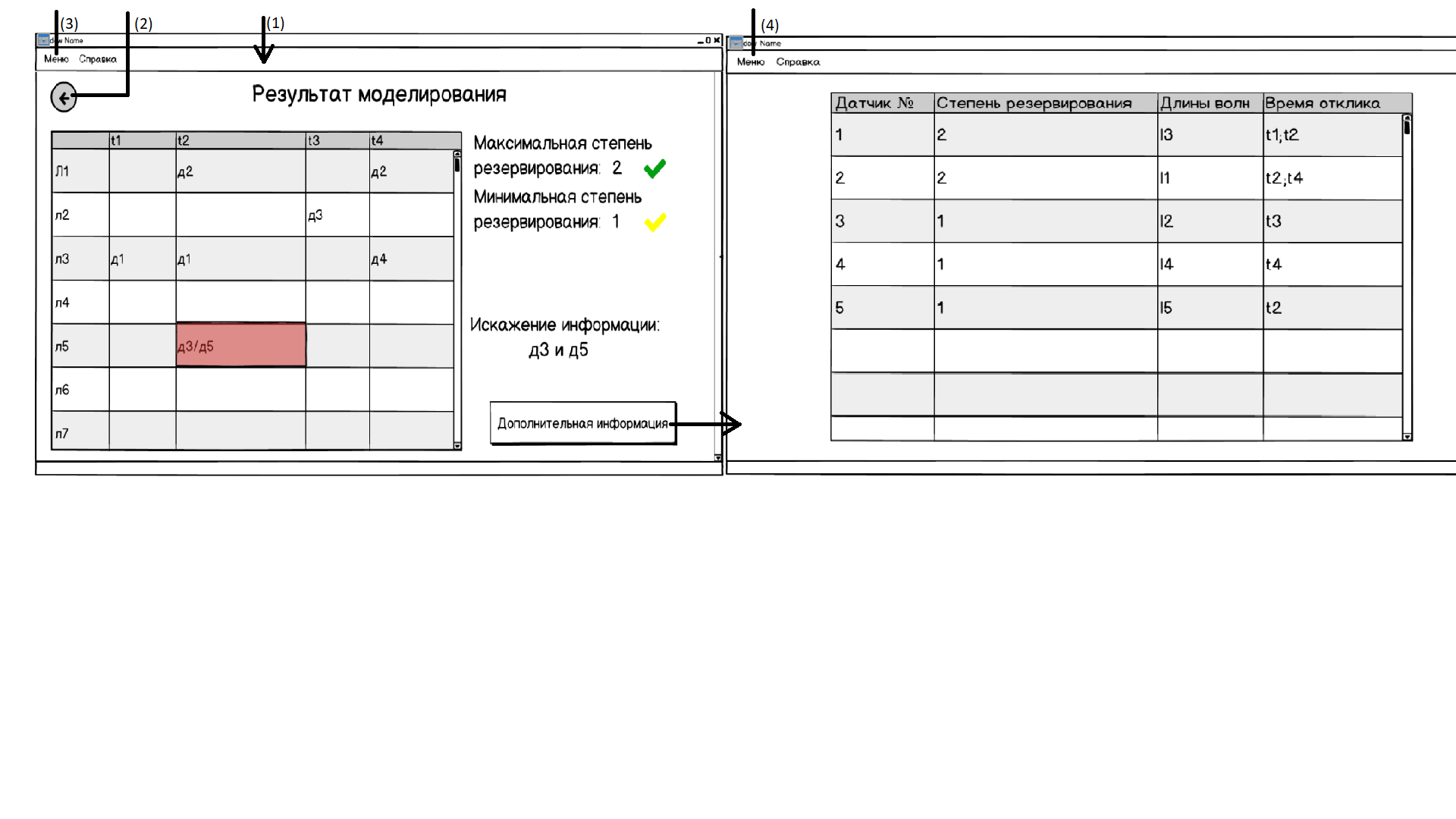


Рисунок ‒ Навигационная модель (окончание)

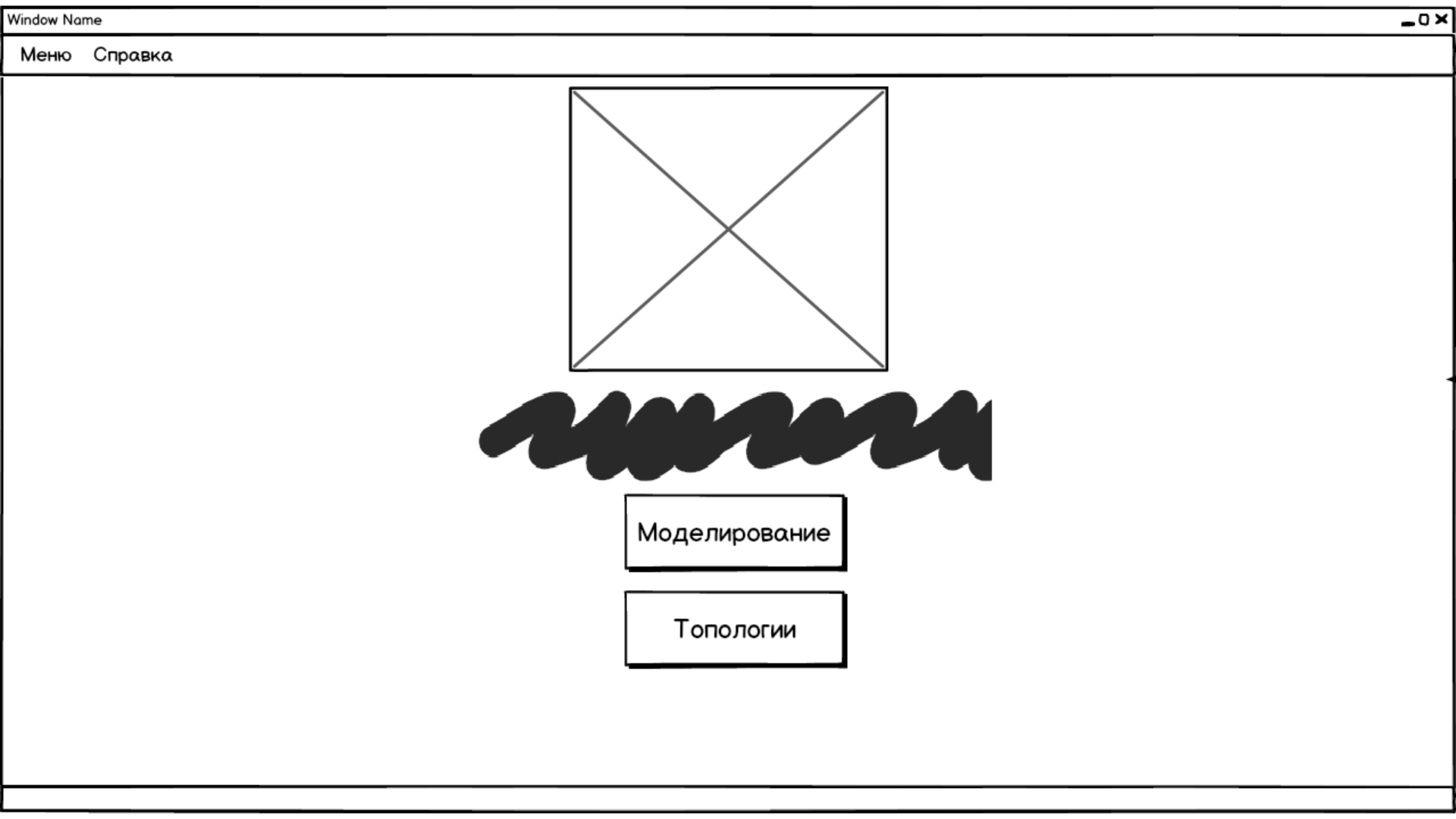


Рисунок ‒ Прототип главной страницы

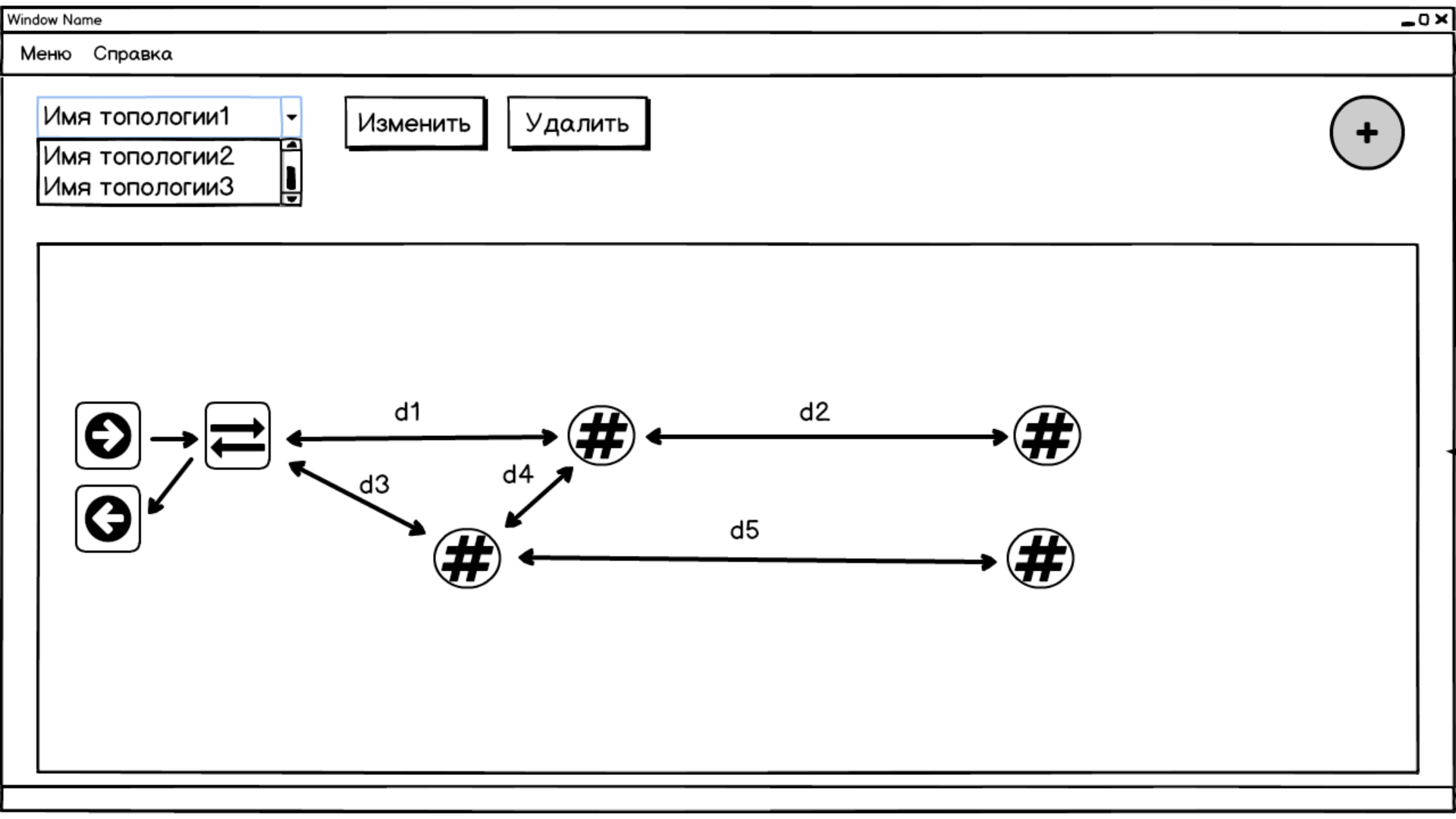


Рисунок ‒ Прототип страницы работы с топологиями

«Удалить» и кнопка с отображением знака «+», что означает создание новой топологии. Пользователь может выбрать топологию, после чего она отоброзится на панели, и удалить, либо изменить ее. При изменении топологии или создании новой, выполнится переход на страницу редактирования, которая представлена на рисунке . Страница должна содержать переключатели для выбора брэгговской решетки и оптоволокна, панель редактирования с возможностью перемещать элементыи указывать им необходимые данные и кнопку «Сохранить». После нажатия кнопки «Сохранить» выполнится переход обратно на страницу работы с топологиями.

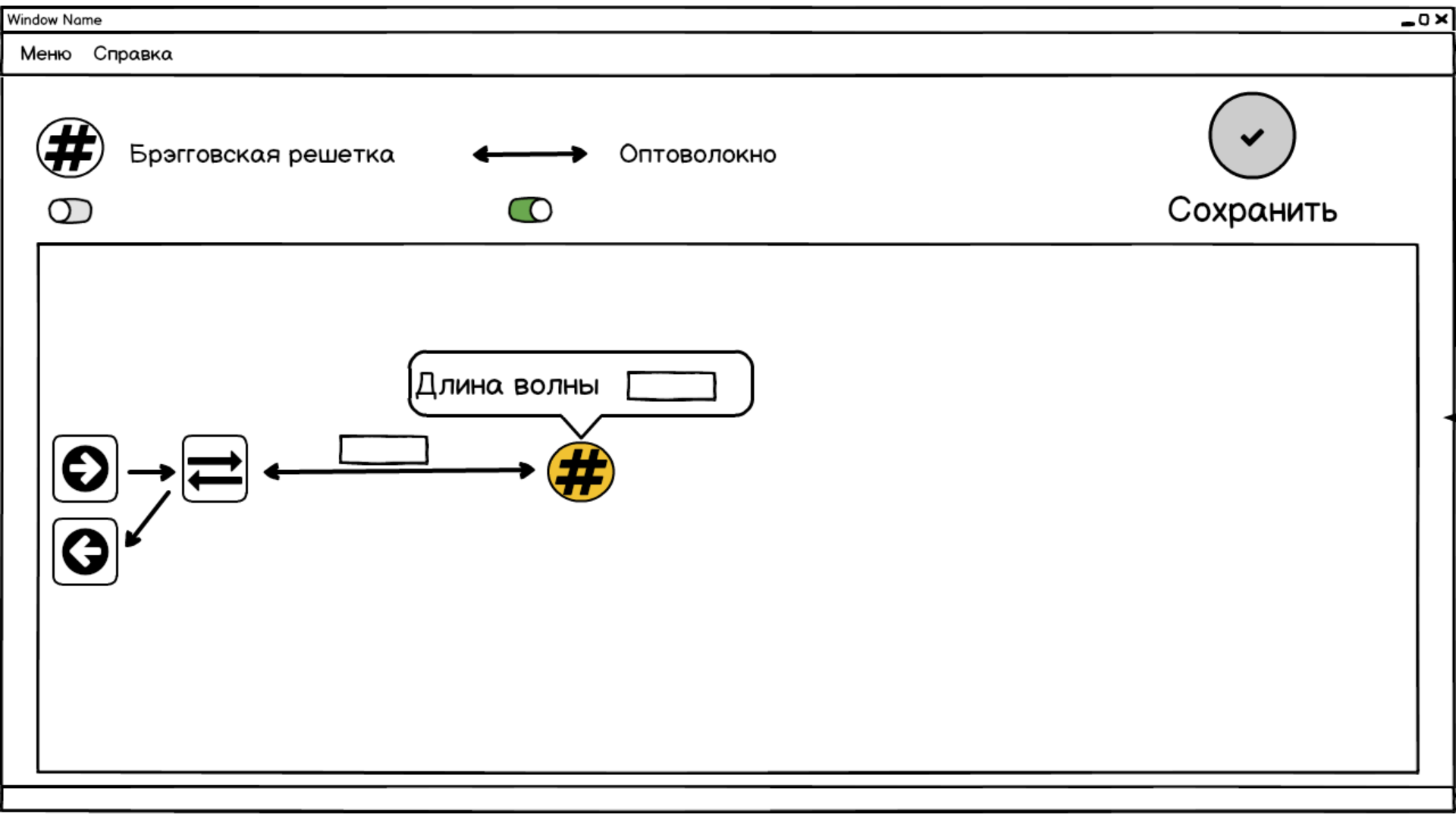


Рисунок ‒ Прототип страницы редактирования топологий

На рисунке представлен прототип страницы моделирования. Она должна содержать список топологий, с возможностью выбора каждой из них, кнопку «+», предназначенную для создания новой топологии, кнопку «Изменить» для изменения выбранной топологии и кнопку «ОК». После нажатия кнопки «ОК» пользователю откроется страница с результатами моделирования, представленная на рисунке .

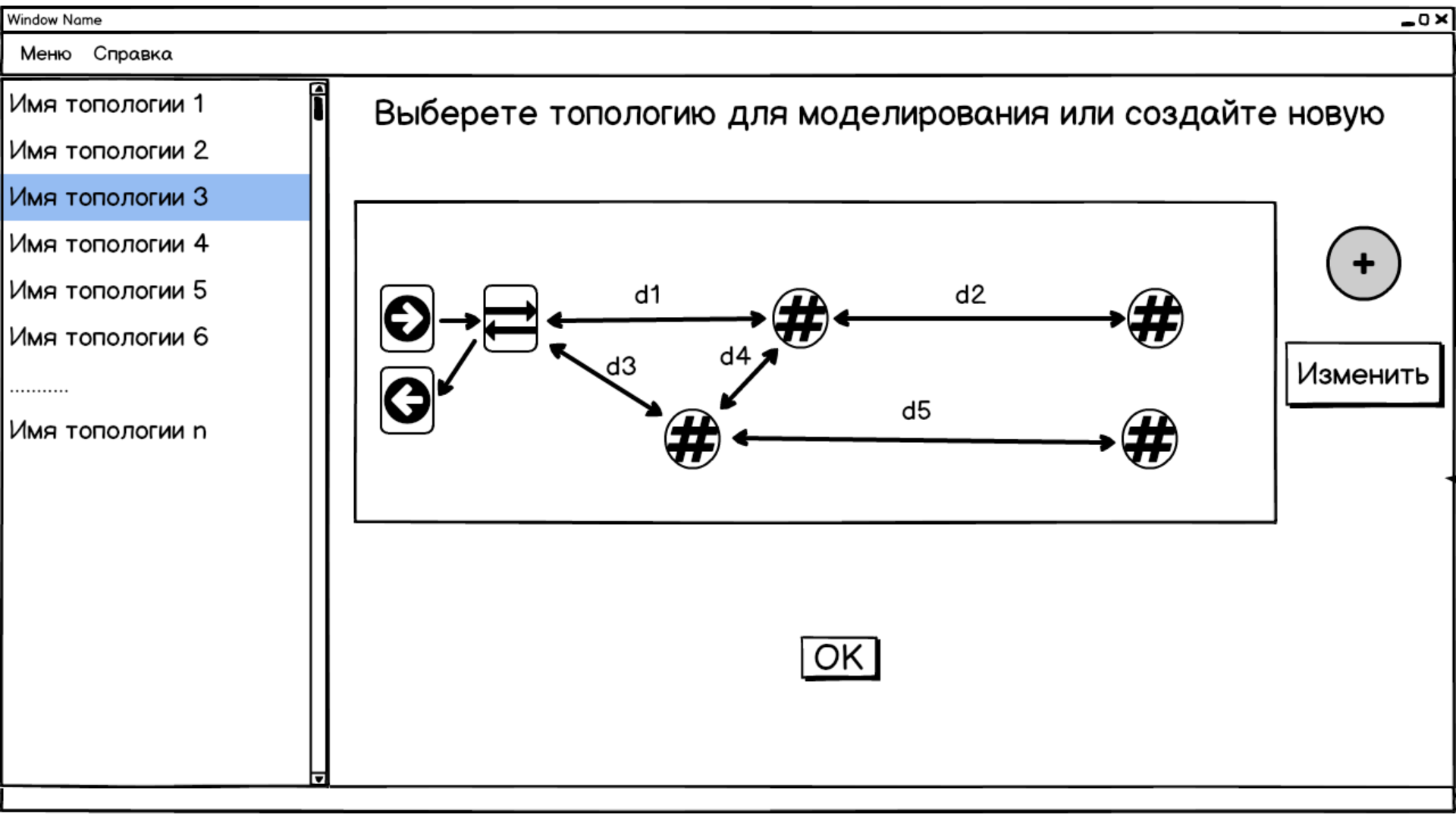


Рисунок ‒ Прототип страницы моделирования



* 1. Информационно-логический проект системы

Унифицированный язык моделирования (UML) является стандартным инструментом для создания "чертежей" программного обеспечения. С помощью UML можно визуализировать, специфицировать, конструировать и документировать артефакты программных систем.

UML пригоден для моделирования любых систем: от информационных систем масштаба предприятия до распределенных Web-приложений и даже встроенных систем реального времени [9].

В нотации языка UML определены следующие виды канонических диаграмм:

* вариантов использования (use case diagram);
* классов (class diagram);
* кооперации (collaboration diagram);
* последовательности (sequence diagram);
* состояний (statechart diagram);
* деятельности (activity diagram);
* компонентов (component diagram);
* развертывания (deployment diagram).
  + 1. Диаграмма вариантов использования

Диаграмма вариантов использования (use case diagram) ‒ это наиболее общее представление функционального назначения системы.

Создание диаграммы вариантов использования имеет следующие цели [9]:

* определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы;
* сформулировать общие требования к функциональному поведению проектируемой системы;
* разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
* подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

*Актер* (actor) ‑ согласованное множество ролей, которые играют внешние сущности по отношению к вариантам использования при взаимодействии с ними (это может быть любой объект, субъект или система, взаимодействующая с моделируемой бизнес-системой извне, т.е. человек, техническое устройство, программа и т.п.).

*Вариант использования* ‑ внешняя спецификация последовательности действий, которые система или другая сущность могут выполнять в процессе взаимодействия с актерами (он определяет набор действий, совершаемый системой при диалоге с актером) [9].

На рисунке показаны варианты использования актером «Пользователь». Пользователь может работать удалять топологии, а так же изменять и добавлять их, указывая все необходимые параметры. Пользователь имеет возможность провести моделирование, выбрав перед этим необходимую топологию. Так же имеется возможность ознакомиться со справкой и переходить в главное меню.

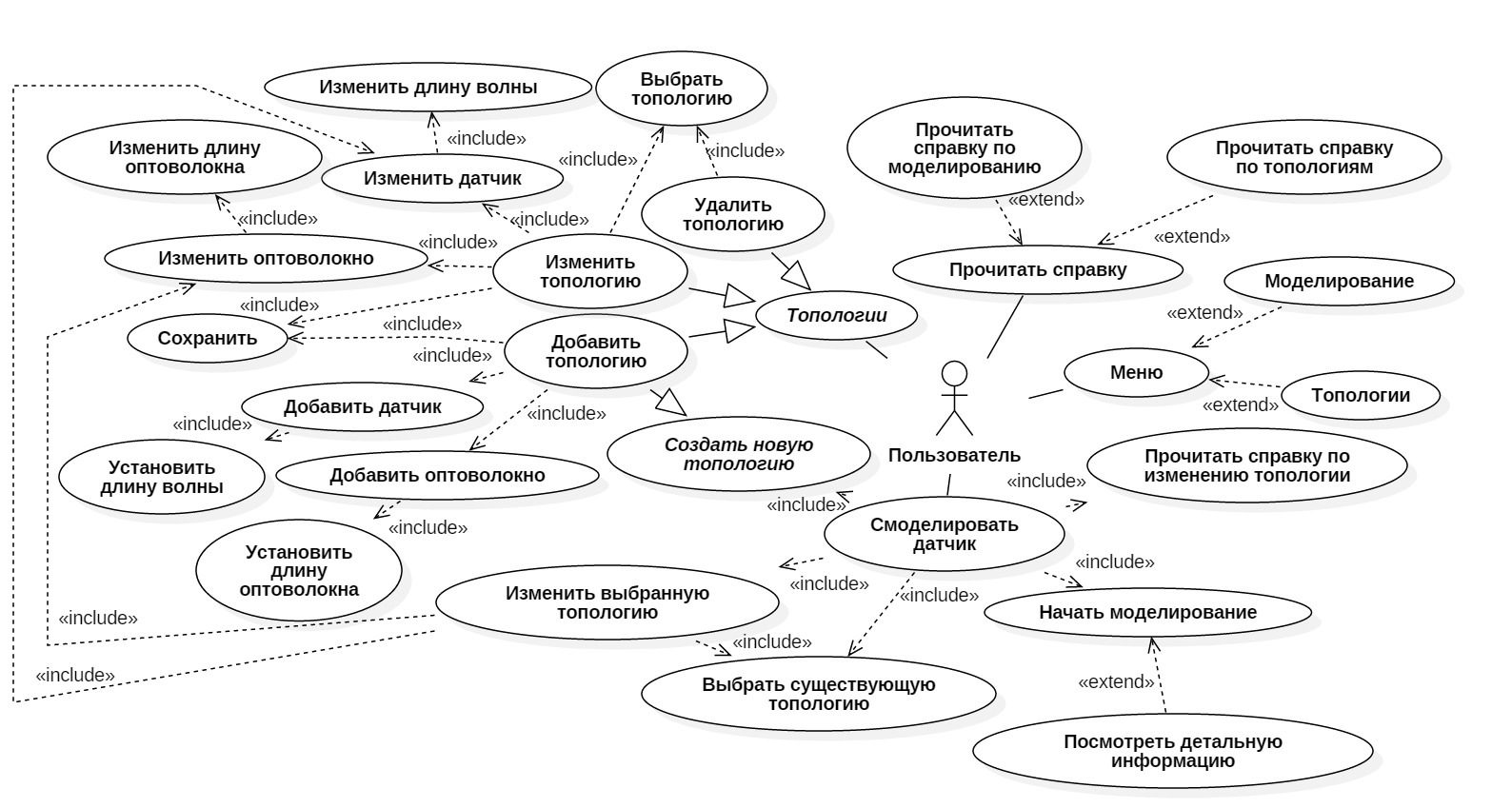


Рисунок ‒ Диаграмма вариантов использования

* + 1. Диаграмма состояний

Диаграммы состояний применяются для того, чтобы объяснить, каким образом работают сложные объекты.

Состояние (state) - ситуация в жизненном цикле объекта, во время которой он удовлетворяет некоторому условию, выполняет определенную деятельность или ожидает какого-то события. Состояние объекта определяется значениями некоторых его атрибутов и присутствием или отсутствием связей с другими объектами.

Диаграмма состояний показывает, как объект переходит из одного состояния в другое, и служат для моделирования динамических аспектов системы [13].

На рисунке изображена диаграмма состояний системы. Запустив программу, система находится в состоянии ожидания действий пользователя. Пользователь может перейти в разделы «Моделирование» и «Топологии», после чего система соединиться с базой данных, запросит список топологий и выведет его на экран. Так же в этих разделах пользователь может создавать и редактировать топологии, переведя систему в состояни редактирования. В этом состоянии пользователь имеет возможность выбирать оптоволокно и оптоволоконную решетку и помещать их на панель редактирования. После нажатия кнопки «Сохранить», система соединиться с базой данных и сохранит новую или обновленную топологию. В разделе «Моделирование» после выбора топологии и нажатия на кнопку «ОК», система перейдет в состояние моделирование, где проводятся математические вычисления. Пользователь может завершить работу с программой путем закрытия окна, нажав на кнопку «Х».

* + 1. Диаграмма деятельности

Диаграмма деятельности (activity diagram) – UML-диаграмма, на которой показано разложение некоторой деятельности на её составные части. Под деятельностью (activity) понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого [14].

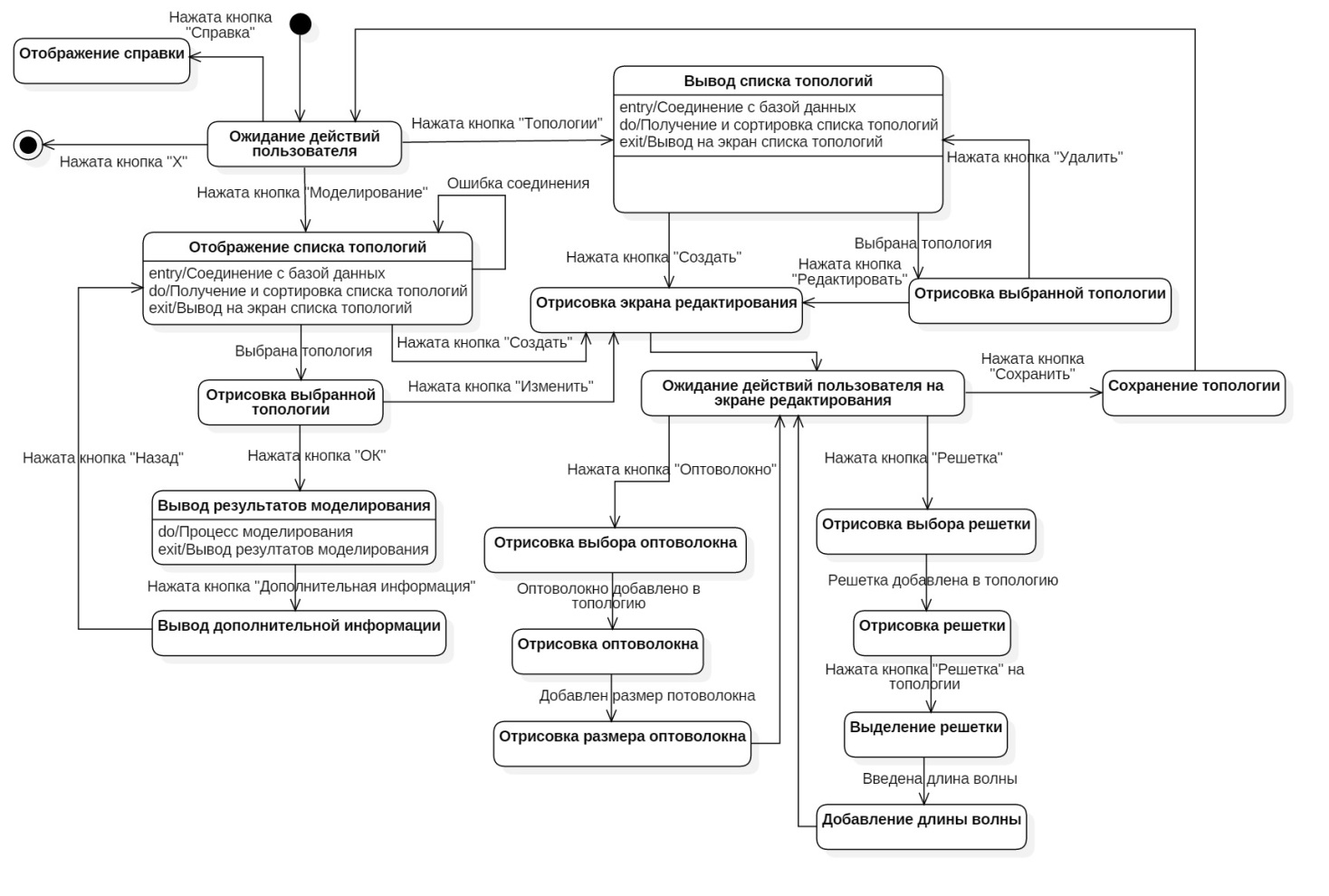


Рисунок ‒ Диаграмма состояний

На рисунке изображена диаграмма деятельности для пользователя. Пользователь имеет возможность в любое время прочитать справку по работе с программой. Если ему необходимо отредактировать список имеющихся топологий, он должен зайти в соответсвующий раздел и либо изменить или удалить топологию, перд этим выбрав необходимую, либо создать новую. В окне редактирования пользователю следует выбирать оптоволокно или оптоволоконную решетку и помещать их на панель редактирования, указав для них необходимые данные. Если же пользователю нужно провести моделирование, он должен перейти в соответсвующий раздел, выбрать топологию и начать моделирование. В результатах моделирования имеется возможность посмотреть дополнительную информацию. Для закрытия программы пользователь должен нажать кнопку «Х».

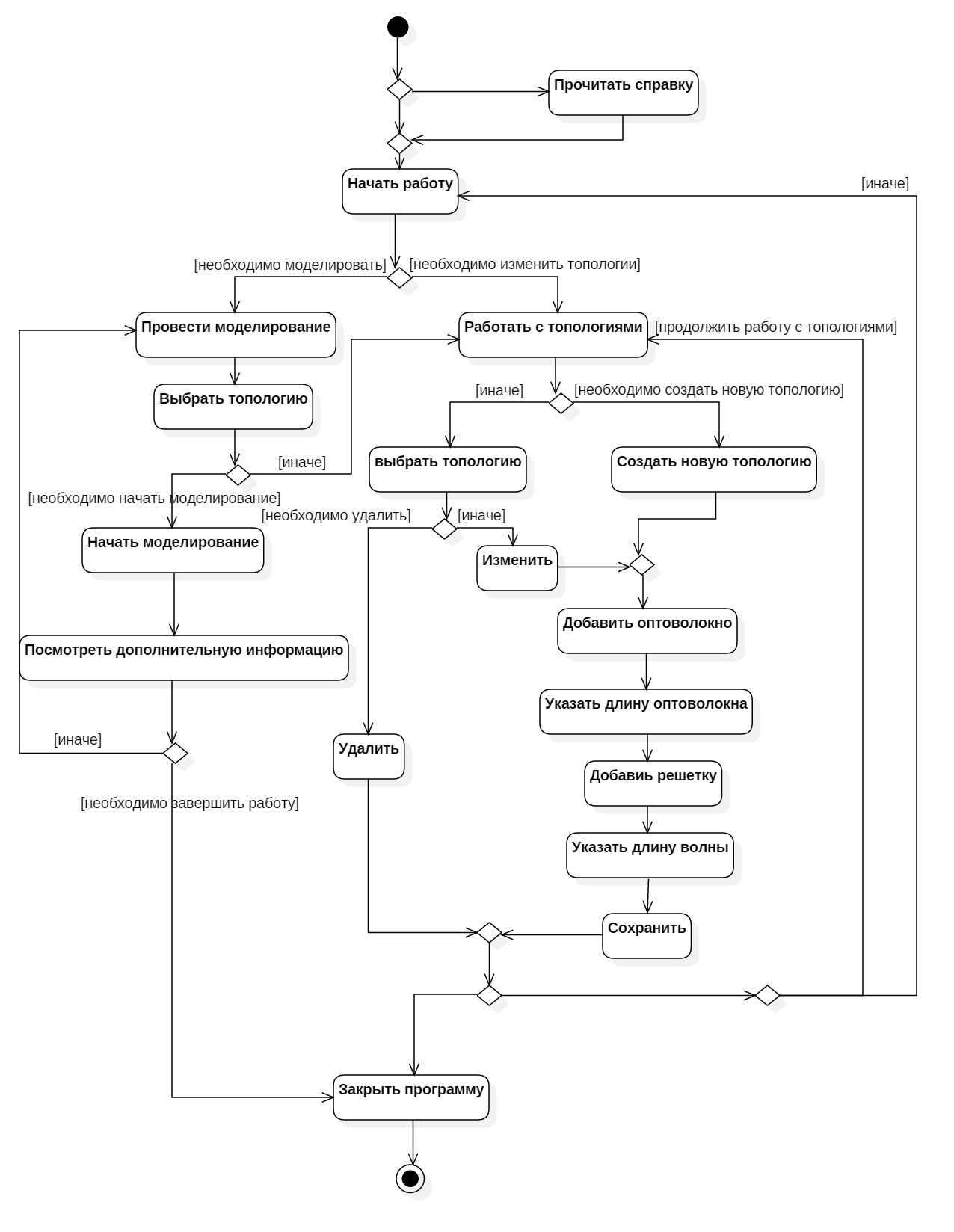


Рисунок ‒ Диаграмма детельности

* + 1. Диаграмма последовательности

Диаграмма последовательности *(sequence diagram)* – диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл (создание-деятельность-уничтожение) и взаимодействие (отправка запросов и получение ответов). Таким образом, данный вид диаграмм отображает взаимодействие объектов в динамике [15].

На рисунке представлена диаграмма последовательности для процесса моделирования датчика от запуска программы до конечного результата.

* + 1. Диаграмма классов

Диаграммы классов – это наиболее часто используемый тип диаграмм, которые создаются при моделировании объектно-ориентированных систем, они показывают набор классов, интерфейсов и коопераций, а также их связи. На практике диаграммы классов применяют для моделирования статического представления системы, они служат основой для целой группы взаимосвязанных диаграмм – диаграмм компонентов и диаграмм размещения  [16].

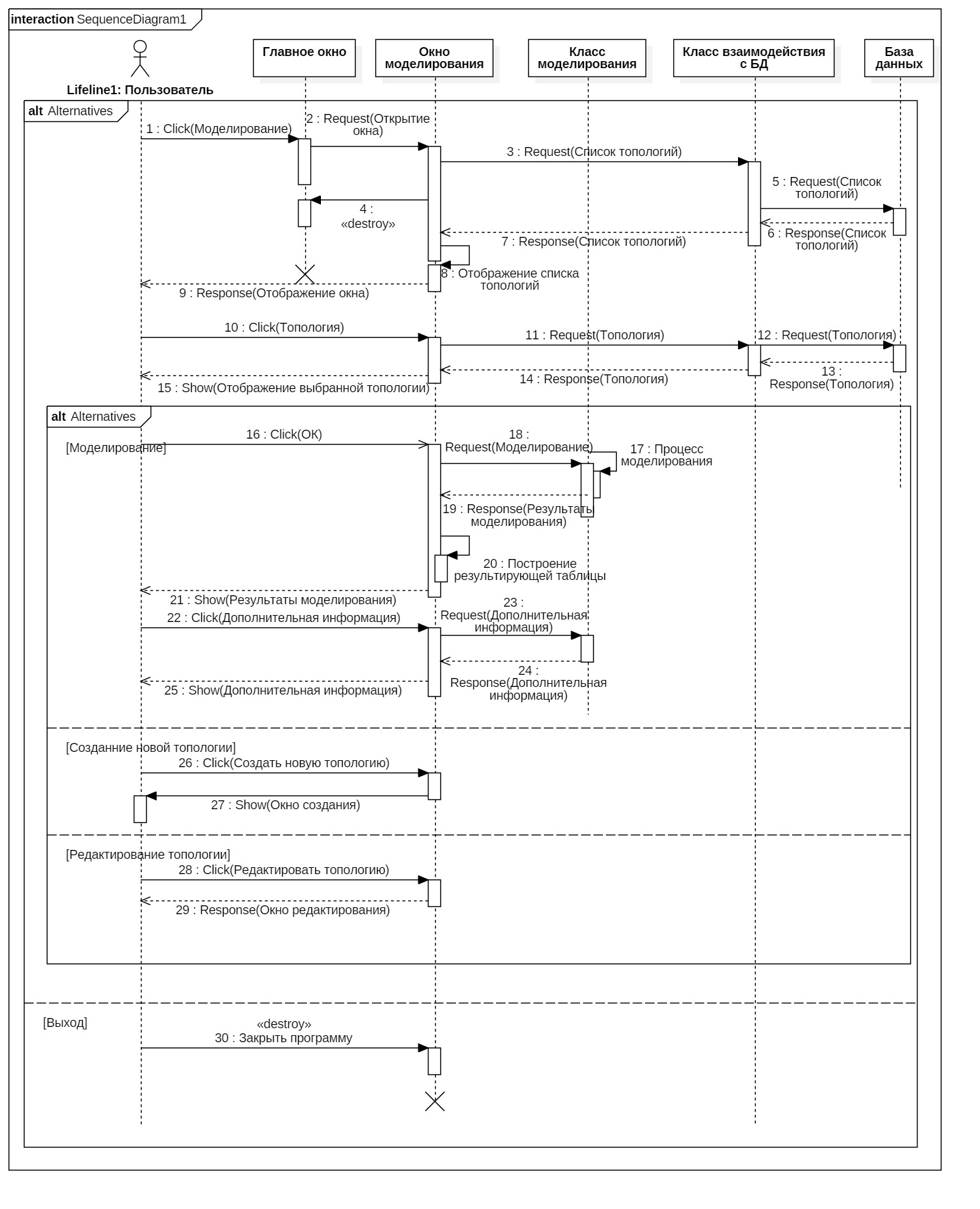


Рисунок ‒ Диаграмма последовательности

* + 1. Логическая модель данных (при необходимости)

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

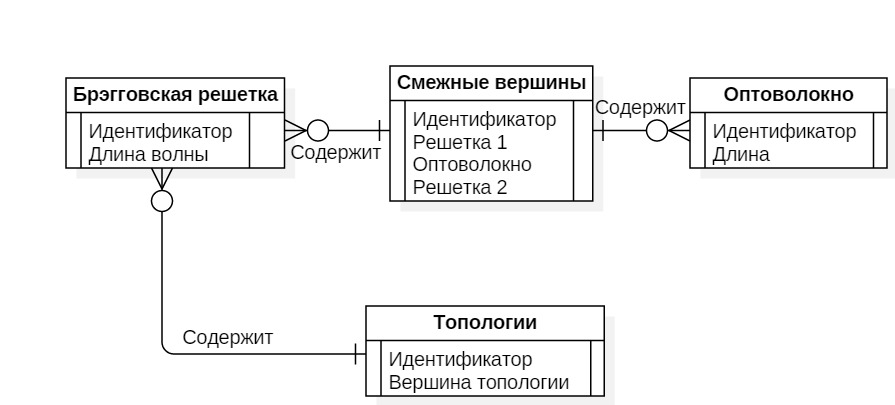


Рисунок ‒ Логическая модель данных

* 1. Выбор и обоснование комплекса программных средств

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор и обоснование операционной системы

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор и обоснование языка программирования

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор и обоснование среды разработки

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор и обоснование СУБД (при необходимости)

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор и обоснование библиотеки ХХХ (при необходимости)

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

1. Реализация системы

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* 1. Описание интерфейса пользователя

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* 1. Диаграммы реализации

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Диаграмма классов

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Диаграмма компонентов

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Диаграмма развертывания

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Физическая модель данных (при необходимости)

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* 1. Выбор и обоснование комплекса технических средств
     1. Расчет объема занимаемой памяти

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Минимальные требования, предъявляемые к системе

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Заключение

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

В первой части приведены Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Во второй части приведены …

Во третьей части приведены …

Результаты работы докладывались на … (перечислить, где участвовали, если есть награды, то тоже указать, в приложении В привести копии грамот, дипломо и т.п.).[[1]](#footnote-1)

Список использованной литературы

1. Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя. Изд. 2-е. М.: ДМК Пресс, 2006. 546 с.
2. Нестационарная аэродинамика баллистического полета/ Липницкий Ю.М. и [др.]. М.: Физматлит, 2003. 176 с.
3. Зеленко Л.С., Шумская Е.А. Комплекс программ для работы с учебным контентом в дистанционных обучающих системах// Известия СНЦ РАН. 2015. №2 (5). Т. 17. С. 992-1003.
4. ГОСТ Р 7.0.4-2006. Издания. Выходные сведения. Общие требования и правила оформления. М., 2006. II. 43 с. (Система стандартов по информ., библ. и изд. делу).
5. Российская гидроэнергетика [Электронный ресурс] // Русгидро: [сайт]. URL: http://www.rushydro.ru/industry/russianhydropower/ (дата обращения: 20.12.2014).
6. Гидроэлектростанция (гидроэлектрическая станция, ГЭС) // Энциклопедический словарь юного техника М.: Издательство «Педагогика», 1987 [Электронный ресурс] // Библиотекарь.Ру: электрон. библ. 2006-2017. URL: http://www.bibliotekar.ru/enc-Tehnika/58.htm (дата обращения: 20.12.2014).
7. Белова С.В. Язык UML. Диаграмма вариантов использования. Систем. требования: PowerPoint. URL: nkse.ru/component/k2/item/  
   download/7\_754f5a247edc6ec6be78218f187338a5.html (дата обращения: 17.03.2017).
8. Философия культуры и философия науки: проблемы и гипотезы: межвуз. сб. науч. тр./ Саратов. гос. ун-т; [под ред. С.Ф. Мартыновича]. Саратов: изд-во Сарат. ун-та, 1999. 199 с.
9. Акимова А.Е., Трешников А.А., Зеленко Л.С. Информационная среда ГЭС. Подсистема расчета показателей эффективности работы оборудования // Перспективные информационные технологии (ПИТ-2017): сб. науч. тр. межд. научно-техн. конф.; [под ред. С.А. Прохорова]. Самара: Изд-во СНЦ РАН, 2017. С. 41-44.
10. Акимова А.Е., Трешников А.А., Зеленко Л.С. Подсистема расчета показателей эффективности работы оборудования // Математика. Компьютер. Образование: труды XXIV межд. конф., 23-28 января 2017 г., г. Пущино. URL: http://www.mce.su/rus/presentations/ p283063/ (дата обращения: 02.03.2017).

Приложение А   
Руководство пользователя

Приложение Б   
Листинг программы

Не менее 8-10 страниц кода, можно 10 кеглем и одинарным интервалом

1. Если есть [↑](#footnote-ref-1)