

Министерство науки и высшего образования Российской Федерации  
 федеральное государственное автономное  
 образовательное учреждение высшего образования  
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники  
 Направление подготовки 09.04.01 Информатика и вычислительная техника  
 Отделение школы (НОЦ) Отделение информационных технологий

### МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
<b>Разработка модуля интеграции для настольного приложения ManyCam компании "ТомскСофт" на основе внутреннего API</b> <b>Integration module development for ManyCam desktop application of TomskSoft company based on the internal API</b>

УДК 004.428:004.43:004.451:004.455.1

Студент

Группа	ФИО	Подпись	Дата
8ВМ01	Видякин Глеб Игоревич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Кочегурова Е. А.	к.т.н., доцент		

Language mentor

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИЯ ИШИТР	Сидоренко Т. В.	к.п.н., доцент		

### КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН ШБИП	Жиронкин С. А.	д.э.н., профессор		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ООД ШБИП	Федоренко О. Ю.	д.м.н., профессор		

### ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП, должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Кочегурова Е. А.	к.т.н., доцент		

**ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП**  
по направлению 09.04.01 Информатика и вычислительная техника

<b>Код компетенции</b>	<b>Наименование компетенции</b>
<b>Универсальные компетенции</b>	
<b>УК(У)-1</b>	Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий
<b>УК(У)-2</b>	Способен управлять проектом на всех этапах его жизненного цикла
<b>УК(У)-3</b>	Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели
<b>УК(У)-4</b>	Способен применять современные коммуникативные технологии, в том числе на иностранном (-ых) языке (-ах), для академического и профессионального взаимодействия
<b>УК(У)-5</b>	Способен анализировать и учитывать разнообразие культур в процессе межкультурного взаимодействия
<b>УК(У)-6</b>	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки
<b>Общепрофессиональные компетенции</b>	
<b>ОПК(У)-1</b>	Способен самостоятельно приобретать, развивать и применять математические, естественно-научные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте
<b>ОПК(У)-2</b>	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач
<b>ОПК(У)-3</b>	Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями
<b>ОПК(У)-4</b>	Способен применять на практике новые научные принципы и методы исследований
<b>ОПК(У)-5</b>	Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем
<b>ОПК(У)-6</b>	Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования
<b>ОПК(У)-7</b>	Способен адаптировать зарубежные комплексы обработки информации и автоматизированного проектирования к нуждам отечественных предприятий
<b>ОПК(У)-8</b>	Способен осуществлять эффективное управление разработкой программных средств и проектов
<b>Профессиональные компетенции</b>	
<b>ПК(У)-1</b>	Способен разрабатывать и администрировать системы управления базами данных
<b>ПК(У)-2</b>	Способен проектировать сложные пользовательские интерфейсы

<b>ПК(У)-3</b>	Способен управлять процессами и проектами по созданию (модификации) информационных ресурсов
<b>ПК(У)-4</b>	Способен осуществлять руководство разработкой комплексных проектов на всех стадиях и этапах выполнения работ
<b>ПК(У)-5</b>	Способен проектировать и организовывать учебный процесс по образовательным программам с использованием современных образовательных технологий

Министерство науки и высшего образования Российской Федерации  
 федеральное государственное автономное  
 образовательное учреждение высшего образования  
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники  
 Направление подготовки (специальность) 09.04.01 Информатика и вычислительная техника  
 Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:  
 Руководитель ООП/ОПОП  
 \_\_\_\_\_ Кочегурова Е.А.  
 (Подпись) (Дата) (Ф.И.О.)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

В форме:

<b>ВКР магистра</b>
---------------------

(ВКР бакалавра/ ВКР специалиста/ ВКР магистра)

Студенту:

Группа	ФИО
8ВМ01	Видякину Глебу Игоревичу

Тема работы:

<b>Разработка модуля интеграции для настольного приложения ManyCam компании "ТомскСофт" на основе внутреннего API Integration module development for ManyCam desktop application of TomskSoft company based on the internal API</b>	
Утверждена приказом директора (дата, номер)	№ 96-20/с от 05.04.2022

Срок сдачи студентом выполненной работы:	01.06.2022
--	------------

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<p><b>Исходные данные к работе</b>  <i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материала изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Объектом проектирования и разработки является модуль интеграции для настольного приложения ManyCam</p>
--	---

<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b></p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> <li>1. Исследование предметной области.</li> <li>2. Разработка архитектуры для разрабатываемого модуля.</li> <li>3. Проектирование и реализация программного модуля.</li> <li>4. Тестирование программного модуля.</li> <li>5. Документирование программного модуля.</li> <li>6. Работа над разделом по финансовому менеджменту, ресурсоэффективности и ресурсосбережения.</li> <li>7. Работа над разделом по социальной ответственности.</li> </ol>
--	---

<p><b>Перечень графического материала</b></p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<ol style="list-style-type: none"> <li>1. Схема архитектуры.</li> <li>2. Диаграммы вариантов использования.</li> <li>3. Диаграмма деятельности.</li> <li>4. Дизайн-макеты интерфейса.</li> </ol>
--	--

<b>Консультанты по разделам выпускной квалификационной работы</b>	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Профессор ОСГН ШБИП, д.э.н., Жиронкин С.А.
Социальная ответственность	Профессор ООД ШБИП, д.м.н., Федоренко О.Ю.
Английский язык	Доцент ОИЯ ИШИТР, к.п.н., Сидоренко Т.В.

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	01.03.2022
---	------------

**Задание выдал руководитель / консультант (при наличии):**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Кочегурова Е.А.	к.т.н., доцент		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ВМ01	Видякин Глеб Игоревич		

Министерство науки и высшего образования Российской Федерации  
 федеральное государственное автономное  
 образовательное учреждение высшего образования  
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники  
 Направление подготовки (специальность) 09.04.01 Информатика и вычислительная техника

Уровень образования Магистратура

Отделение школы (НОЦ) Отделение информационных технологий

Период выполнения \_\_\_\_\_ (осенний / весенний семестр 2021 /2022 учебного года)

Форма представления работы:

ВКР магистра
--------------

(ВКР бакалавра/ ВКР специалиста/ ВКР магистра)

### КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Обучающегося:

Группа	ФИО
8ВМ01	Видякина Глеба Игоревича

Тема работы:

<b>Разработка модуля интеграции для настольного приложения ManyCam компании "ТомскСофт" на основе внутреннего API Integration module development for ManyCam desktop application of TomskSoft company based on the internal API</b>
---

Срок сдачи студентом выполненной работы:	01.06.2021
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
01.06.2021	Исследование предметной области	20
01.06.2021	Разработка архитектуры для разрабатываемого модуля	20
01.06.2021	Проектирование и реализация программного модуля	20
01.06.2021	Тестирование программного модуля	10
01.06.2021	Документирование программного модуля	10
01.06.2021	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
01.06.2021	Социальная ответственность	10

**СОСТАВИЛ:**

**Руководитель ВКР**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Кочегурова Е.А.	к.т.н., доцент		

**СОГЛАСОВАНО:**

**Руководитель ООП**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Кочегурова Е.А.	к.т.н., доцент		

**Обучающийся**

Группа	ФИО	Подпись	Дата
8ВМ01	Видякин Глеб Игоревич		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа		ФИО	
8ВМ01		Видякину Глебу Игоревичу	
Школа	Инженерная школа информационных технологий и робототехники	Отделение школы (НОЦ)	Отделение информационных технологий
Уровень образования	Магистр	Направление/специальность	09.04.01 Информатика и вычислительная техника

### Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Бюджет проекта – не более 250000 руб., в т.ч. затраты по оплате труда – не более 150000 руб. Стоимость материальных и интеллектуальных ресурсов определялась согласно прейскурантам компаний Оклад студента-программиста – 25000 р.
2. <i>Нормы и нормативы расходования ресурсов</i>	Значение показателя интегральной ресурсоэффективности – не менее ... баллов из ... Накладные расходы 16 %; Районный коэффициент 30%; Норма амортизации ПЭВМ 33,33%; Норма амортизации ПО 20%
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Коэффициент отчислений на выплаты во внебюджетные фонды 30%

### Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого и инновационного потенциала НТИ</i>	<i>Анализ потенциальных потребителей результатов исследования; Анализ конкурентных технических решений; FAST анализ; SWOT анализ; Оценка готовности проекта к коммерциализации</i>
2. <i>Разработка устава научно-технического проекта</i>	<i>Цели и результатов проекта; Организационная структура проекта;</i>
3. <i>Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок</i>	<i>Структура работ; План проекта; Бюджет научного исследования; Риски проекта;</i>
4. <i>Определение ресурсной, финансовой, экономической эффективности</i>	<i>Абсолютная эффективность исследования; Сравнительная эффективность исследования;</i>

### Перечень графического материала (с точным указанием обязательных чертежей):

1. «Портрет» потребителя результатов НТИ
2. Сегментирование рынка
3. Оценка конкурентоспособности технических решений
4. Матрица SWOT
5. График проведения и бюджет НТИ
6. Оценка ресурсной, финансовой и экономической эффективности НТИ
7. Потенциальные риски

Дата выдачи задания для раздела по линейному графику

### Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН ШБИП ТПУ	Жиронкин Сергей Александрович	д-р. экон. наук		

### Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ01	Видякин Глеб Игоревич		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
8ВМ01		Видякину Глебу Игоревичу	
Школа	Инженерная школа информационных технологий и робототехники	Отделение (НОЦ)	Отделение информационных технологий
Уровень образования	магистратура	Направление/специальность	<b>09.04.01 Информатика и вычислительная техника</b>

Тема ВКР:

<b>Разработка модуля интеграции для настольного приложения ManyCam компании "ТомскСофт" на основе внутреннего API Integration module development for ManyCam desktop application of TomskSoft company based on the internal API</b>	
<b>Исходные данные к разделу «Социальная ответственность»:</b>	
<p><b>Введение</b></p> <ul style="list-style-type: none"> <li>– Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения.</li> <li>– Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации</li> </ul>	<p>Объект исследования: Модуль интеграции для настольного приложения «ManyCam»          Область применения: информационные системы          Рабочая зона: офис          Размеры помещения :6*8м          Технические параметры помещения:          Центральное отопление, естественная непринудительная вентиляция, освещение искусственное в виде люминесцентных ламп и естественное (2 окна размером 1,5*2м)          Количество и наименование оборудования рабочей зоны: персональные компьютер в количестве 3 штук          Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне:          Разработка приложения с использованием компьютера</p>
<b>Перечень вопросов, подлежащих исследованию, проектированию и разработке:</b>	
<p><b>1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения</b></p> <ul style="list-style-type: none"> <li>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</li> <li>– организационные мероприятия при компоновке рабочей зоны.</li> </ul>	<ul style="list-style-type: none"> <li>– Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 09.03.2021);</li> <li>– ГОСТ 12.0.003-2015 Опасные и вредные производственные факторы. Классификация. Перечень опасных и вредных факторов.</li> <li>– СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания,</li> <li>– СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95,</li> <li>– ГОСТ 12.1.003-83 Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности.</li> <li>– Рабочее место при выполнении работ в положении сидя должно соответствовать требованиям ГОСТ 12.2.032-78.</li> <li>– ТОИ Р-45-084-01 «Типовая инструкция по охране труда при работе на персональном компьютере»,</li> <li>– ГОСТ 12.1.030-81 Система стандартов безопасности труда (ССБТ). Электробезопасность. Защитное заземление.</li> </ul>

	<p>Зануление.</p> <ul style="list-style-type: none"> <li>– ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов,</li> <li>– ГОСТ 12.1.004-91 Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Общие требования/</li> <li>– ГОСТ 17.4.3.04-85 Охрана природы (ССОП). Почвы. Общие требования к контролю и охране от загрязнения/</li> </ul>
<p><b>2. Производственная безопасность при разработке проектного решения:</b></p> <ul style="list-style-type: none"> <li>– Анализ выявленных вредных и опасных производственных факторов</li> <li>– Расчет уровня опасного или вредного производственного фактора</li> </ul>	<p>Вредные факторы:</p> <ul style="list-style-type: none"> <li>– недостаточная освещённость рабочей зоны; отсутствие или недостаток естественного света;</li> <li>– повышенный уровень шума;</li> <li>– повышенный уровень электромагнитных излучений;</li> <li>– повышенная напряжённость электрического поля;</li> <li>– повышенная или пониженная влажность воздуха;</li> <li>– статические перегрузки;</li> </ul> <p>Опасные факторы:</p> <ul style="list-style-type: none"> <li>– электрический ток (источником является ПК);</li> <li>– короткое замыкание;</li> <li>– статическое электричество;</li> </ul> <p>Требуемые средства коллективной и индивидуальной защиты от выявленных факторов: наушники, устройства для вентиляции и очистки воздуха, источники света, устройства улавливания и очистки воздуха и жидкостей;</p> <p>Расчет: повышенный уровень шума</p>
<p><b>3. Экологическая безопасность при разработке проектного решения</b></p>	<p>Воздействие объекта на атмосферу, гидросферу, селитебную зону не происходит. В работе проведён анализ воздействия на литосферу (образование отходов при выходе из строя ПК, возникновения отходов при печати и утилизации ламп).</p>
<p><b>4. Безопасность в чрезвычайных ситуациях при разработке проектного решения</b></p>	<p>Возможные ЧС: Природные катастрофы, например, ураган Геологические воздействия (землетрясения, оползни, обвалы, провалы территории и т.д.); Техногенные аварии (пожар) Наиболее типичная ЧС: пожар</p>
Дата выдачи задания для раздела по линейному графику	

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ООД ШБИП	Федоренко Ольга Юрьевна	Доктор медицинских наук		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ВМ01	Видякин Глеб Игоревич		

## РЕФЕРАТ

Выпускная квалификационная работа 97 с., 22 рис., 27 табл., 38 источников.

**Ключевые слова:** JavaScript, WebSocket, SDK, идеоконференция, модуль интеграции, библиотека, модульное тестирование.

**Объектом исследования является** модуль для интеграции функций API программного комплекса ManyCam в сторонние настольные и веб-приложения.

**Цель работы:** спроектировать и разработать модуль для интеграции функций API программного комплекса ManyCam в сторонние настольные и веб-приложения.

**В процессе исследования проводился** анализ предметной области, проектирование модуля.

**В результате исследования** разработан, протестирован, задокументирован модуль для программного комплекса ManyCam компании «ТомскСофт».

**Степень внедрения:** внедрен, находится в общем доступе.

**Область применения:** заказчики компании «ТомскСофт».

**Экономическая эффективность/значимость работы:** разработка является экономически эффективной.

**В будущем планируется** расширение функциональности созданного модуля и его сопровождение.

## Table of contents

Introduction .....	15
List of definitions .....	17
1. Subject area analysis .....	19
1.1. ManyCam application .....	20
1.2. API и SDK .....	22
1.3. Software development tools choice .....	23
1.3.1. WebSocket .....	24
1.3.2. JavaScript .....	25
1.3.3. Node.js and NPM .....	26
1.3.4. Webpack .....	26
2. SDK design .....	28
2.1. Architecture of system components interaction .....	28
2.2. Functional requiremnts .....	29
2.3. Use cases .....	31
2.4. Activity and Sequence Diagrams .....	32
3. SDK development .....	35
3.1. Analysis of similar implementations .....	35
3.2. Project setup .....	35
3.3. Project build .....	37
3.4. Project structure .....	39
3.5. Main methods .....	42
3.6. Unit testing .....	44
3.7. Documentation .....	47
3.8. Web publication .....	50

4.	SDK usage for development .....	52
4.1.	SDK setup .....	52
4.2.	Demo Application .....	52
5.	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	58
	Введение.....	58
5.1.	Оценка коммерческого и инновационного потенциала НТИ ...	58
5.1.1.	Потенциальные потребители результатов исследования ...	59
5.1.2.	Анализ конкурентных технических решений.....	60
5.1.3.	SWOT анализ.....	60
5.1.4.	Оценка готовности проекта к коммерциализации .....	63
5.2.	Инициация проекта.....	65
5.2.1.	Цели и результат проекта.....	65
5.2.2.	Организационная структура проекта.....	66
5.2.3.	Ограничения и допущения.....	66
5.3.	Планирование управления НТИ.....	67
5.3.1.	Контрольные события проекта.....	67
5.3.2.	План проекта .....	68
5.3.3.	Бюджет НТИ.....	70
5.3.4.	Риски проекта .....	74
5.4.	Определение ресурсной, финансовой, экономической эффективности.....	75
	Вывод .....	77
6.	Социальная ответственность .....	78
	Введение.....	78

6.1. Правовые и организационные вопросы обеспечения безопасности .....	78
6.1.1. Особенности законодательного регулирования проектных решений.....	78
6.1.2. Организационные мероприятия при компоновке рабочей зоны.....	79
6.2. Производственная безопасность .....	80
6.2.1. Недостаточная освещённость рабочей зоны; отсутствие или недостаток естественного света;.....	82
6.2.2. Повышенный уровень шума .....	82
6.2.3. Повышенный уровень электромагнитных излучений .....	84
6.2.4. Повышенная напряжённость электрического поля; .....	84
6.2.5. Повышенная или пониженная влажность воздуха;.....	85
6.2.6. Статические перегрузки;.....	85
6.2.7. Электробезопасность .....	87
6.2.8. Статическое электричество.....	88
6.3. Экологическая безопасность .....	88
6.3.1. Воздействие на литосферу .....	88
6.4. Безопасность в чрезвычайных ситуациях .....	90
6.4.1. Пожарная безопасность.....	90
Выводы.....	92
Conclusion .....	93
Bibliography.....	94

## **Introduction**

Nowadays video communication is becoming one of the main ways of interaction between people. The most popular of it is video conferencing, which is used for:

- distant study;
- distant work;
- emergency situations;
- reducing travel expenses in geographically distributed organizations;
- improving the efficiency of remote communication;
- telemedicine, etc.

Video communication software uses a limited set of video sources, among them are a webcam and screen sharing. However, there are many other sources, such as: IP cameras, presentations, games and applications, statistical images and others. This problem is solved by third-party enterprise software. One such software package is ManyCam. This is an application developed by TomskSoft. It allows users, conducting remote lectures and broadcasting on streaming platforms, to apply video conferencing for upgrading video streams.

ManyCam has a user interface (UI) that suits for most user's needs. However, some customers need to embed ManyCam features into their software programs. For this reason, the application is personally integrated into other systems. Such integration is made by TomskSoft developers, who makes the API. It takes much time and resources.

In order to solve this problem, it was decided to create a documented open-source SDK, i.e., a set of software development tools in one package. The SDK will allow other developers to independently create applications based on the internal API of ManyCam software.

The tasks of this work are to create an SDK for the software application ManyCam, unit testing, documentation and further support.

The purpose of this work is to design and develop integration module for ManyCam API functions that can be implemented into third-party desktop and web applications.

## List of definitions

API (Application Programming Interface) – a description of the ways (a set of classes, procedures, functions, structures or constants) in which one computer program can interact with another program [1].

CRUD (create, read, update, delete) – an acronym for the four basic functions used when working with data.

CSS (Cascading Style Sheets) – a formal language for describing the markup of web documents [3].

GET – method that is used to request the content of the specified resource.

Git – a free and open-source distributed version control system designed to handle any project, from small to very large, quickly and efficiently [27].

GitHub – is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuous integration, and wikis for every project [6].

HTML (HyperText Markup Language) – standardized markup language for documents on the World Wide Web [7].

HTTP (HyperText Transfer Protocol) – is an application layer protocol in the Internet protocol suite model for distributed, collaborative, hypermedia information systems [8].

JavaScript (JS) – is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS [5].

NPM (Node Package Manager) – package manager included with Node.js.

Node.js – is a JavaScript runtime built on Chrome's V8 JavaScript engine [10].

POST – a method that is used to pass user data to a given resource.

SDK (Software Development Kit) – is a set of software development tools for communication with API in one installable package [11].

UI (User Interface) – it is the way a user interacts with a product, service, or service.

Web Application – application software that runs on a web browser, unlike software programs that run locally and natively on the operating system (OS) of the device. Web applications are delivered on the World Wide Web to users with an active network connection [14].

Web server – a server that accepts HTTP requests from clients, usually web browsers, and provides them with HTTP responses, usually along with an HTML page, image, file, media stream, or other data.

OOP (object-oriented programming) – programming methodology based on representing a program as a set of objects, each of which is an instance of a certain class, and classes form an inheritance hierarchy [33].

Фреймворк – set of templates for the software platform that define the structure of the software system.

Singleton pattern – a creational design pattern that lets you ensure that a class has only one instance, while providing a global access point to this instance [13].

Builder pattern – a creational design pattern that lets you construct complex objects step by step. The pattern allows you to produce different types and representations of an object using the same construction code [2].

## **1. Subject area analysis**

A video conference is a communication session between two users or a group of users, regardless of their location.

Various peripheral equipment is used to provide participants with sound and video: cameras, screens, microphones, headsets. Modern video and audio codecs, specialized network protocols, various signal processing algorithms make it possible to achieve high-quality communication on almost any communication channel.

Videoconferencing is used as a tool to make decisions in various situations: in emergency situations; to reduce travel expenses in geographically distributed organizations; to improve communication efficiency; to conduct trials with remote participation of convicts; as one of the elements of telemedicine and distance learning technologies.

In many government and commercial organizations, video conferencing brings great results and maximum efficiency, for example:

- reduces travel time and associated costs;
- accelerates decision-making processes in emergency situations;
- reduces the time for consideration of cases in courts;
- increases general work productivity;
- resolves personnel issues and social situations;
- allows you to make more informed decisions by bringing in additional experts as needed.

Often, during a video conference, it is necessary to demonstrate various media data. In this case video conferencing software allows user to share screen or its individual windows, but usually nothing more. However, there are many other video sources such as IP cameras, presentations, games and applications, statistical images and others. This problem is solved by third-party enterprise software. One of such programs is the ManyCam system developed by TomskSoft.

## 1.1. ManyCam application

ManyCam is the go-to live streaming software and virtual camera that allows you to deliver professional live videos on any streaming platform, video conferencing app, and distance learning tool. This software allows to use multiple cameras and other video sources such as mobile devices, presentations, IP cameras, games. Also, user has opportunity to use virtual backgrounds, add new video layers and scenes, broadcast the desktop and much more.

The following describes the ManyCam application, its main functions and how it works.

ManyCam application user interface is shown on Figure 1.

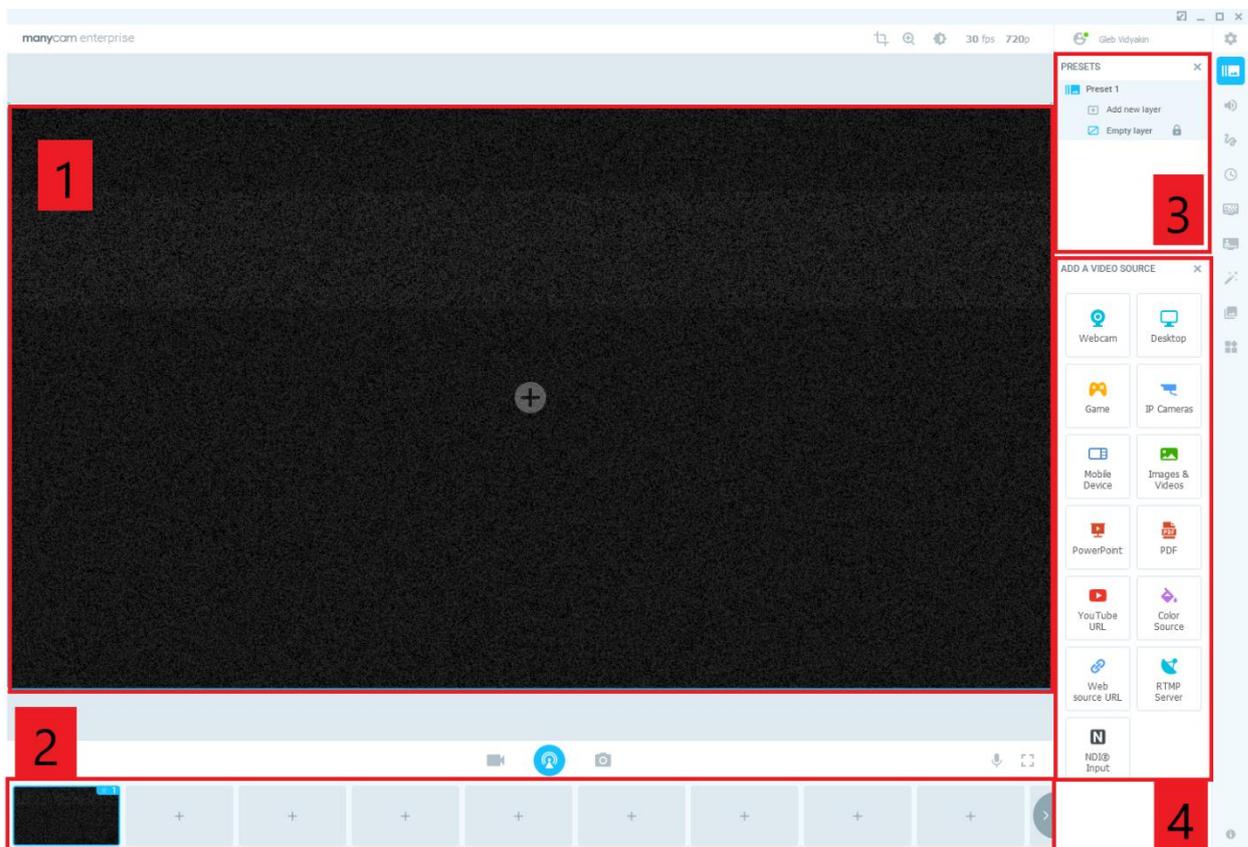


Figure 1 – ManyCam main window

There are 4 red squares on this figure:

1. Video output (preset).
2. Panel for creating and selecting scenes (presets).
3. Control panel for layers in presets.

#### 4. Video source selection panel.

The maximum number of presets is 24. Presets can contain the number of layers, limited by system resources. The more layers, the more CPU, GPU, RAM are loaded. User has an opportunity to load one of completely different resources into the layer. It is allowed to use following video sources:

1. Webcam.
2. Display screen.
3. Video game.
4. IP camera.
5. Mobile device.
6. Media files.
7. Power Point presentation.
8. PDF file.
9. Video from the Internet.
10. Filling the background.
11. RTMP Server.
12. NDI input.

Therefore, user can create preset that fits his needs. An example of a preset with different layers is shown in Figure 2.

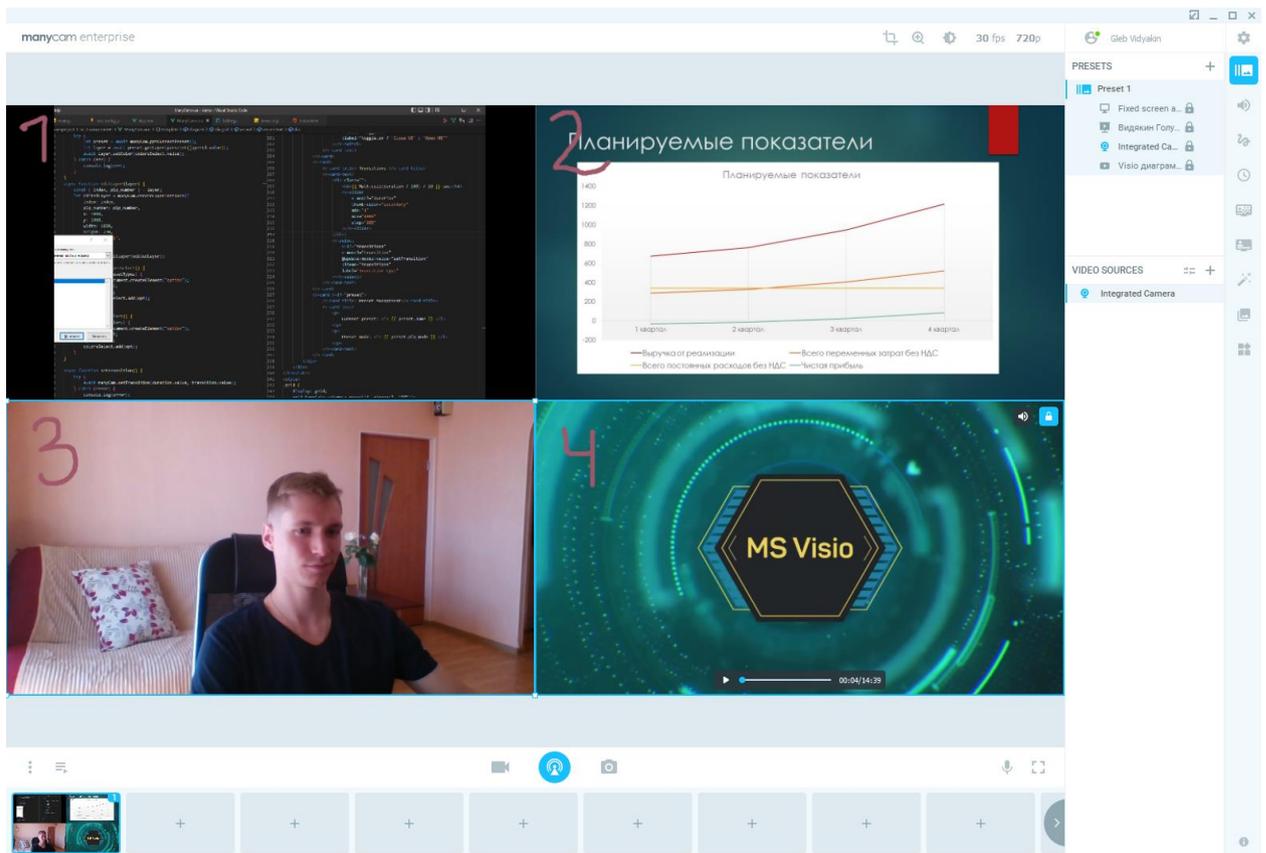


Figure 2 – Layers example

Following layers are numbered in the figure:

1. Translation of the application window.
2. PowerPoint presentation.
3. Webcam.
4. YouTube video.

Further, in order to use the resulting video, for example, in Zoom, user needs to select «ManyCam Virtual Webcam» from the list of cameras in application.

## 1.2. API и SDK

API (application programming interface) is a set of protocols and tools that provide data exchange between different components of information systems [12].

When the software application provides an API, then third-party developers have the opportunity to use and embed the modules of this application into new software.

SDK [11] (software development kit) is a set of software development tools in one installable library. The SDK solves a larger problem: not only to provide data exchange between the application and third-party infrastructure, but to implement a full-fledged process. It can contain components for receiving user data, secure processing and storage, changing states [12].

The SDK usually contains supporting code, extensive documentation. Also, it can use several APIs. This is not only an interface for working with the system, but a complete set of tools for implementing some business logic. The SDK interacts with the application's API and is a layer between the third-party application and the API. Companies create SDKs in order to make development much easier for third-party developers. It allows them to solve their problems through abstraction not diving into the code.

SDKs are mainly used for developing applications on a specific platform. For example, developing Android applications that run on the Java platform requires a Java Development Kit; for iOS applications – iOS SDK; for the Universal Windows Platform – .NET Framework SDK.

### **1.3. Software development tools choice**

The ManyCam software application is a desktop application for the end user. That means that relationships between TomskSoft company and clients described as Business to Consumer (B2C). The program has a graphical interface that suits the needs of most users. However, some customers need to embed ManyCam features into their applications. Thus, the relationships between the client and the customer become Business to Business (B2B) level.

For this reason, the application is personally integrated into other systems. Such integration requires developers of the company who are engaged in API development. Integration takes a long time and consumes company resources. To solve this problem, it was decided to create an open-source documented SDK.

Implemented SDK should match following requirements:

1. ManyCam provides an API based on the WebSocket protocol. This protocol enables persistent connection and two-way communication between ManyCam and third-party applications. Therefore, the chosen programming language must have support for the WebSocket protocol.
2. Basically, most of integrations are done primarily in web applications, that is why chosen technology stack must be web-oriented.
3. In addition, support in desktop applications is required.

To fulfil these requirements, it was decided to use JavaScript programming language. This choice is based on following factors:

1. JavaScript has the most widespread use in web applications.
2. It is possible to develop desktop applications using JavaScript.
3. The language has native browser support for the WebSocket protocol.
4. There are convenient ways of publishing implemented solution in the Internet.

Node.js is required for building and testing the SDK.

GitHub will be used to control and facilitate development.

Versions will be published under the MIT license in the free NPM package manager.

Selected development tools will be explained in next sections.

### **1.3.1. WebSocket**

The WebSocket protocol provides the ability to exchange data between a browser and a server over a persistent connection. Data is transmitted through it in both directions in the form of packets, without breaking the connection and additional HTTP requests [9].

There is a big amount of control commands from the client and events from the server. This also requires a duplex connection. For this reason, the WebSocket protocol was chosen.

To open a web socket connection in JavaScript, it is needed to create a WebSocket object by specifying the ws url:

```
let socket = new WebSocket("ws://*url*");
```

Once the WebSocket object is created, you can listen for its events. There are only 4 of them:

1. Open – connection is established.
2. Message – receive server's answer.
3. Error – receive server's answer error.
4. Close – the connection is closed.

After the object is created, it starts to establish a connection.

To establish it, client and the server make a handshake over the HTTP protocol. After the connection is established, client and server start working over the WebSocket protocol, which is no longer HTTP (Figure 3).

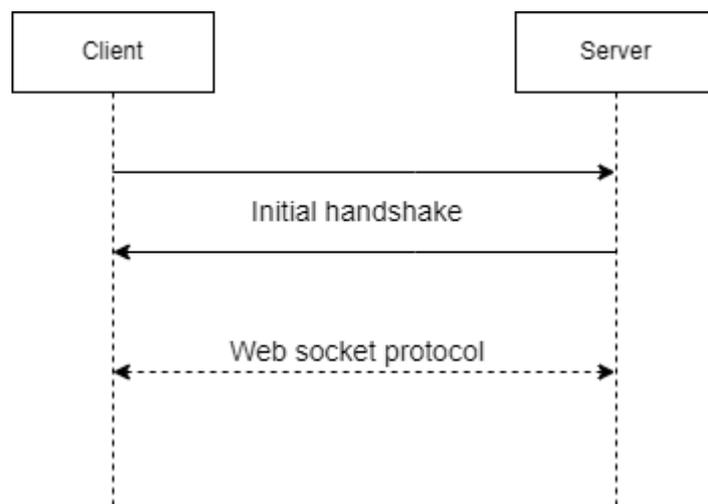


Figure 3 – WebSocket protocol

### 1.3.2. JavaScript

JavaScript [29] is a high-level, often just-in-time compiled language. It has dynamic typing and prototype-based object-orientation.

Structurally, JavaScript can be represented as a union of three clearly distinguishable parts:

- core (ECMAScript);

- Browser Object Model or BOM;
- Document Object Model or DOM.

The browser object model and document object model are not supported in non-browser JavaScript environments.

Only the browser object model supports the WebSocket protocol. Therefore, to develop desktop applications, it's required to install special libraries that implement WebSocket protocol. This is one of the reasons for choosing this programming language. Another important factor is the widespread use of the language in web development.

### **1.3.3. Node.js and NPM**

Node.js [30] is a JavaScript language interpreter. Node.js is a C++ application that takes JavaScript as input and executes it.

A Node.js package is one or more JavaScript files that represent some kind of library or tool.

NPM (node package manager) is a standard package manager that is automatically installed with Node.js. It is used to download packages from the npm cloud server, or to upload packages to these servers.

Once NPM initiated in the project there is the package.json file in root project directory. The package.json file contains information about the application: name, version, dependencies, etc. Any directory that contains this file is interpreted as a Node.js package.

### **1.3.4. Webpack**

For building a project, it is important to have a project builder. It automates the routine tasks of building a project into a single file. For these purposes, Webpack was chosen, as it is a powerful and flexible tool that allows to compile JavaScript modules into a single JS file.

With a large number of files, it creates a single file (or several files) to run the application.

In this case, Webpack is used to create two builds: one for the browser and one for the server, as each runtime is different.

## 2. SDK design

An important step in software development is the design of the future software architecture. It does not matter if it is a library or a large distributed system – design is necessary for the successful implementation and development of the project.

The right system architecture allows to achieve required development goals: availability, fault tolerance, scalability, ease of maintenance and support.

### 2.1. Architecture of system components interaction

The architecture of ManyCam software is shown in Figure 4.

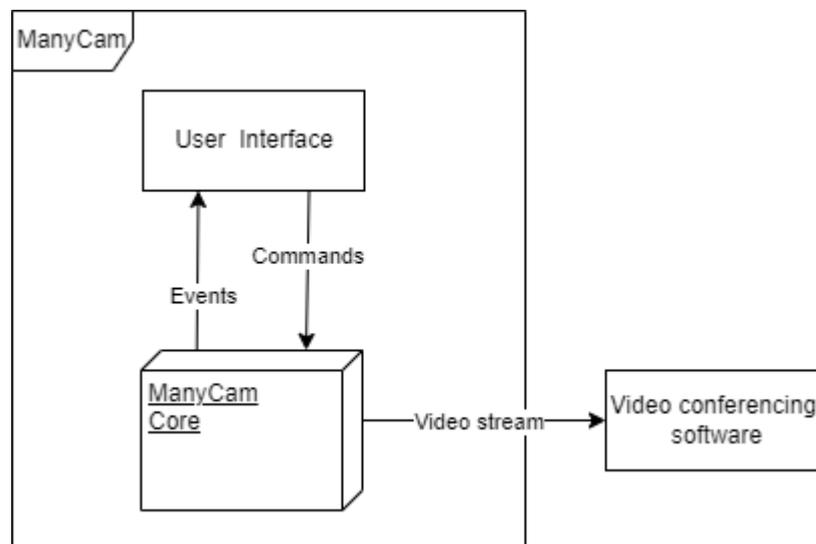


Figure 4 – ManyCam architecture

The system contains a kernel that processes user interface commands and signals back to the interface when certain events occur. The system processes all video sources defined by the user and generates a video stream that can be used in video conferences.

Also, for integration into other systems, ManyCam provides a programming interface through which it is possible to interact with the application. Communication is based on the WebSocket protocol.

The SDK implements an internal API and provides a set of tools for developing a third-party application. The complete architecture of the software package is shown in Figure 5.

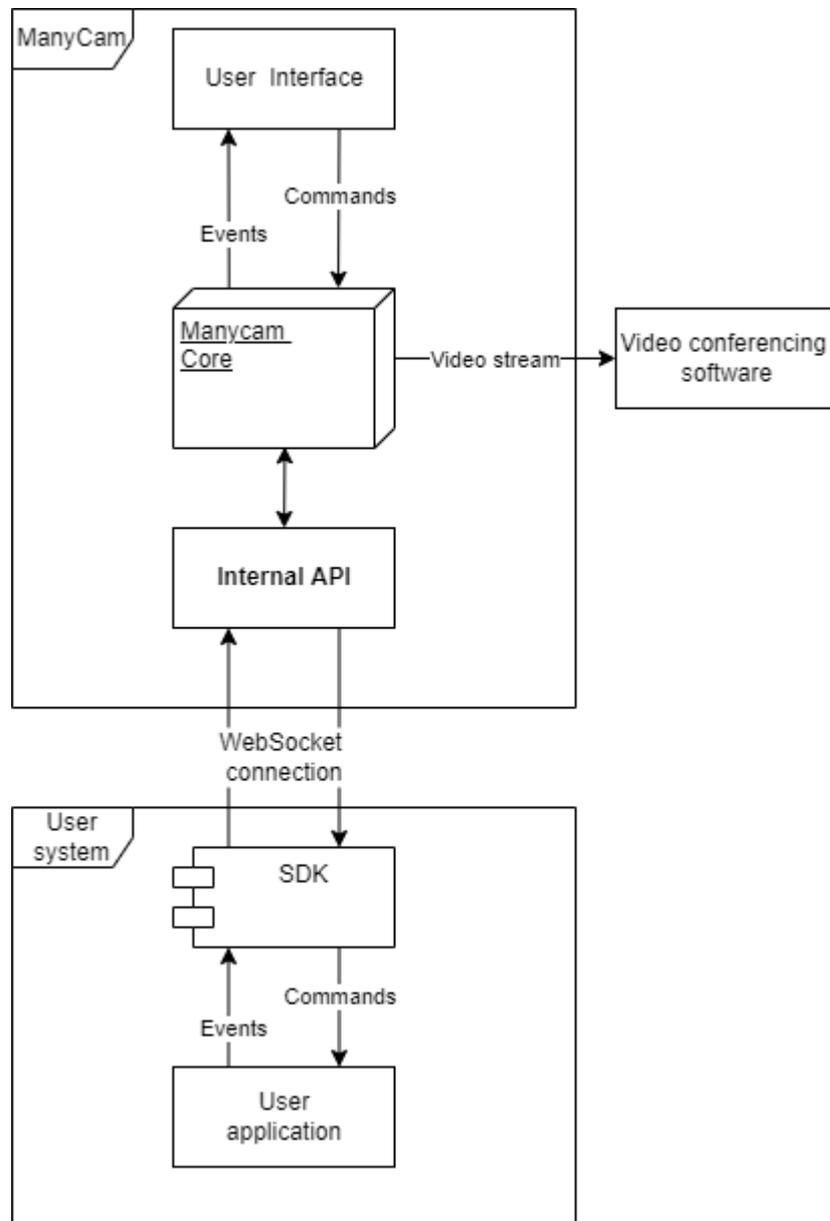


Figure 5 – Full system architecture

## 2.2. Functional requirements

The ManyCam software can be used by users with and without an activated subscription. In the second case, the set of application features is limited. Therefore, the internal API has the same limitations. Thereby, 2 roles were created: user and user with a subscription.

Functional requirements are divided into general and user with subscription requirements.

General requirements:

1. Ability to connect to the API.
2. Ability to log in.
3. Ability to log out of the system.
4. Exit the application.
5. Get the state of the application (hidden / open).
6. Show application window.
7. Hide the application window.
8. Get a list of available cameras.
9. Get a list of available resources.
10. Get the current preset.
11. Get layer preset.
12. Make the preset active.
13. Make the layer active.

User with subscription requirements:

1. Create a preset.
2. Edit preset.
3. Delete preset.
4. Clear preset.
5. Get the state of the preset.
6. Get a list of available video resolutions.
7. Get current video resolution.
8. Set permission.
9. Create a screenshot.
10. Create layer.
11. Edit layer.
12. Remove layer.
13. Listen to an event that fires if the preset has been edited.

14. Listen to an event that fires if the preset has been changed.
15. Listen to an event that fires when the subscription changes.
16. Listen to an event that fires on log in/log out.
17. Listen to an event that is triggered if motion is detected on the preset.
18. Listen to an event that fires if an error occurs.

### 2.3. Use cases

Based on the functional requirements, a use case diagrams were built (Figure 6, Figure 7).

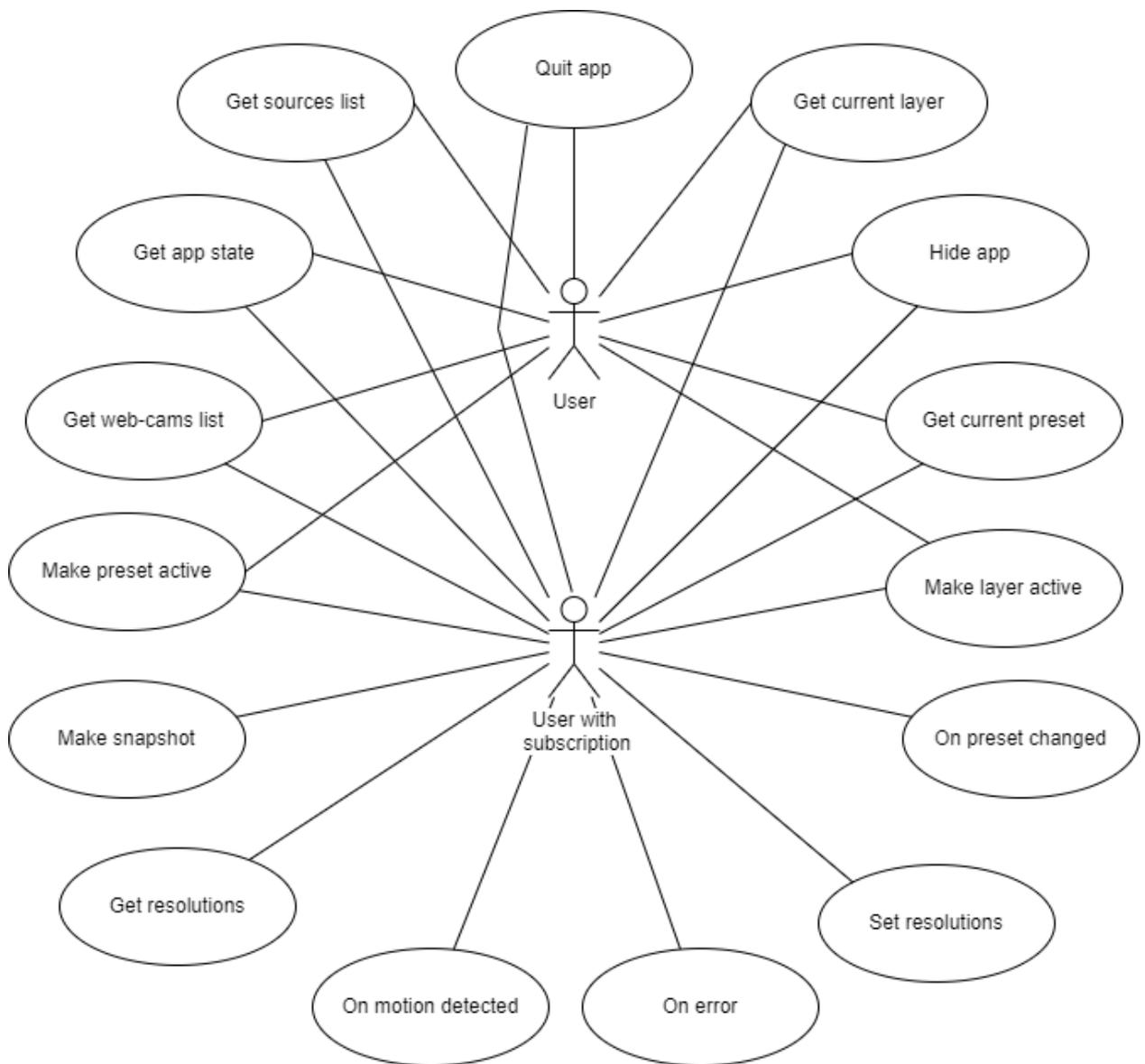


Figure 6 – Use case diagram: general and for user with subscription

A user and a user with subscription share some functionality. All users can get lists of video resources to output, application states, current preset and layer, as well as switch between layers and presets, hide the application and exit it and etc.

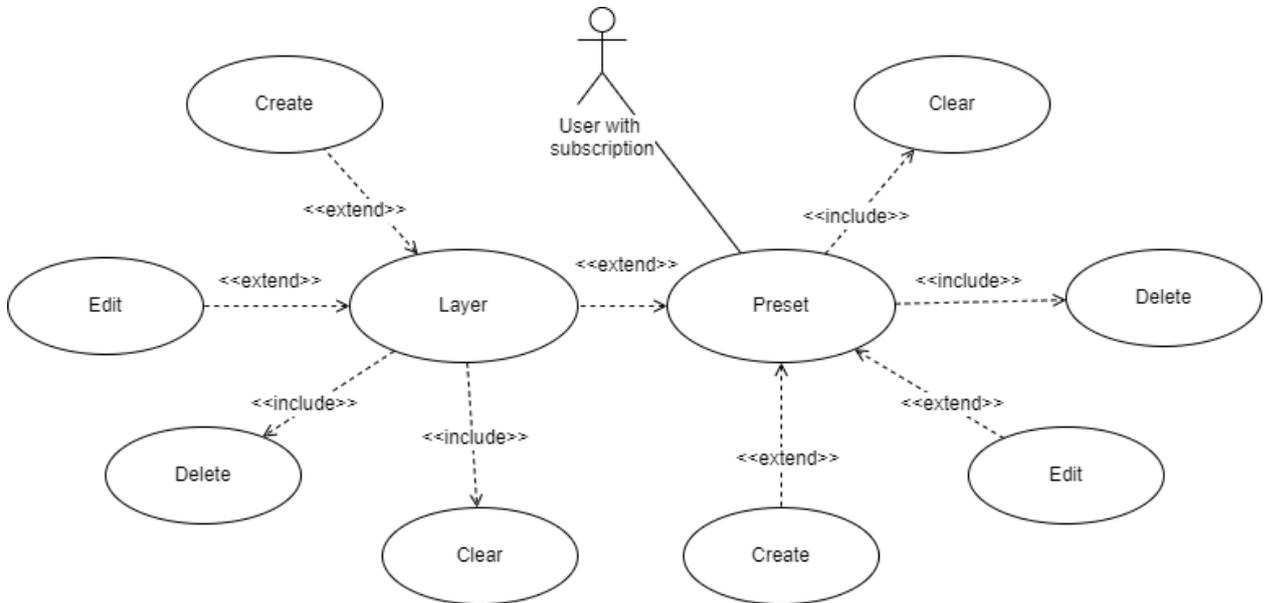


Figure 7 – Use case diagram: Preset and Layer CRUD operations

A subscribed user has access to all general functions, CRUD operations with presets and layers, all kind of events, take screenshots, get all kind of lists.

## 2.4. Activity and Sequence Diagrams

There is an activity diagram [26] that represents communication between third-party application using SDK and ManyCam software (Figure 8).

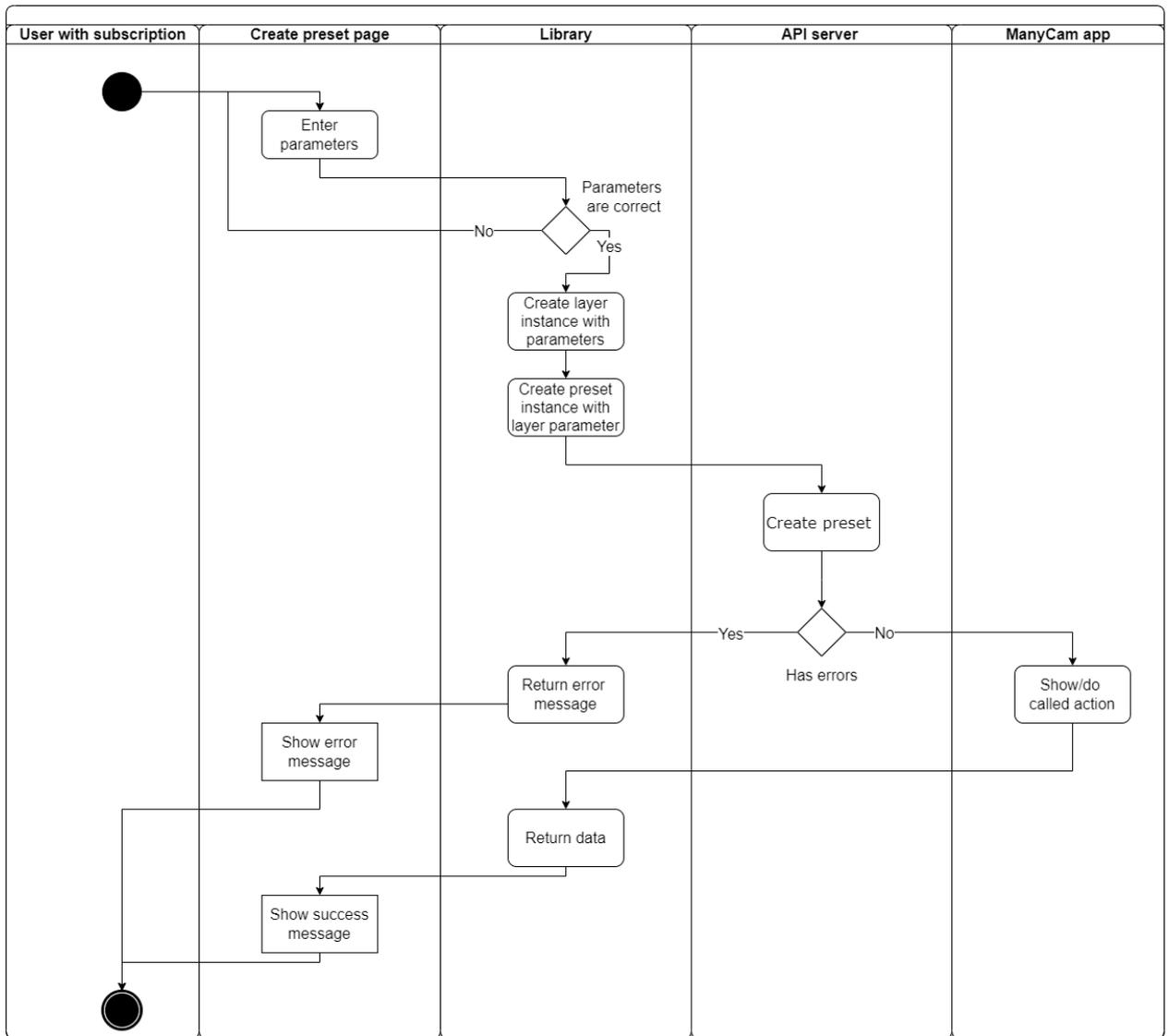


Figure 8 – Activity diagram

The diagram shows the process of creating a preset using the SDK library.

The process activator is the user with the subscription. First, user enters parameters for future preset: preset parameters (type, number of presets) and layer parameters (video sources, video resolution, crop, rotate, size and position). The SDK checks parameters for the correctness and if all parameters are valid it creates Preset object with layers. Then using this object SDK make and API query to create a preset. Next SDK gets data with creation status and process it. Finally, user gets a notification if creation is successful or not.

The sequence diagram (Figure 9) describes the process of a user subscribing to the "Movement detected" event.

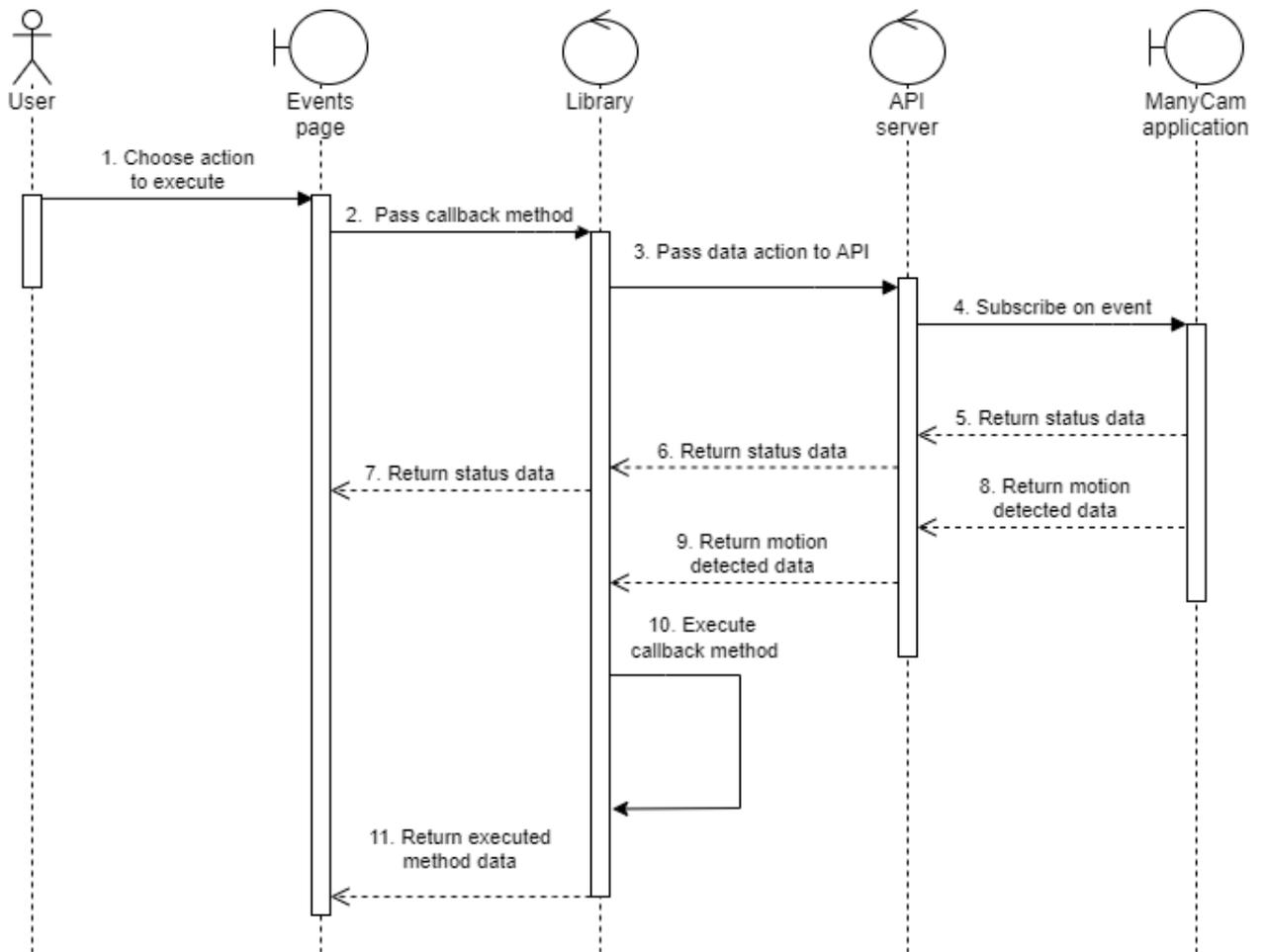


Figure 9 – Sequence diagram

After the user initiates process third-party application subscribe to the library event with callback function. This function will be called after the event fires. Then SDK subscribes to API event. SDK subscribes on application event and returns status data to SDK then to the user. API server listens to the application events and when it fires return event's data to API server then to the SDK. SDK executes callback function and return its data to the user. It's a cyclical process – application will listen to the event until the user unsubscribes or connection is lost.

### **3. SDK development**

Before creating the SDK, similar projects on GitHub were explored. There are some architectural implementations and solutions that fit this work.

#### **3.1. Analysis of similar implementations**

There are two implementations on GitHub that use same technologies:

1. Zoom Video SDK is a software module for embedding the functions of the Zoom software package into web applications.
2. Kuzzle SDK is a software module that uses the WebSocket protocol.

Both libraries use the singleton design pattern: one instance of the class is used to call methods, because it is necessary to maintain a connection with the API server.

The libraries also use asynchronous function calls (Promises). This is an important part of the implementation, since promises are needed so that code execution does not stop when requests are made to the server.

The Kuzzle SDK library is based on the WebSocket protocol, the code provides examples of connection, error and event handling.

Although the applications use similar development tools, they are not analogous to the SDK under development.

Further development was carried out taking into account the above features.

#### **3.2. Project setup**

The public repository on GitHub has been initialized.

Next, npm was installed and configured. When the project was initialized, the package.json file was created (Figure 10).

```
package.json > ...
...
1  {
2    "name": "manycam-sdk",
3    "version": "1.0.0",
4    "description": "SDK for ManyCam Websocket API",
5    "main": "index.js",
6    "keywords": [
7      "manycam",
8      "js",
9      "websocket",
10     "node",
11     "browser"
12  ],
13  "scripts": {
14    "buildProd": "webpack --mode=production",
15    "buildDev": "webpack --mode=development",
16    "test": "jest ./test --noStackTrace",
17    "doc": "jsdoc -c jsdoc.json -d docs/html",
18    "docMD": "node createDocs.js"
19  },
20  "author": "Gleb Vidyakin",
21  "license": "MIT",
22  "exclude": [
23    "node_modules"
24  ],
25  "dependencies": {
26    "md5": "^2.3.0",
27    "ws": "^7.4.3"
28  },
29  "devDependencies": {
30    "@babel/core": "^7.12.17",
31    "@babel/plugin-proposal-class-properties": "^7.12.13",
32    "@babel/plugin-transform-runtime": "^7.12.17",
33    "@babel/preset-env": "^7.12.17",
34    "@babel/runtime": "^7.12.18",
35    "@types/jest": "^26.0.20",
36    "babel-loader": "^8.2.2",
37    "bufferutil": "^4.0.3",
38    "clean-webpack-plugin": "^3.0.0",
39    "jest": "^26.6.3",
40    "jsdoc-to-markdown": "^7.0.0",
41    "utf-8-validate": "^5.0.4",
42    "webpack": "^5.23.0",
43    "webpack-cli": "^4.5.0",
44    "webpack-merge": "^5.7.3"
45  }
46 }
```

Figure 10 – Package.json file

It specifies:

1. Title, project description, version, keywords, author's name and license type – it will be used for publishing the package repository project and visible to all users.
2. The entry point for the building is the main file, which contains all modules of the library.
3. Scripts – functions for building, testing and documentation.
4. Project dependencies – other libraries for SDK development.

After initialization and during development, the listed dependencies were installed. The dependency directory (node\_modules) is not indexed by version control.

### 3.3. Project build

The Webpack module is used to build the project. Building the project is necessary for minifying the program code and transpiling it to support older versions of the language.

According to the requirements, it is needed to make sure that the SDK can work both on the server side (from Node.js) and on the client side (from the browser).

The problem is that the browser has native support for the WebSocket protocol. And in Node.js, support is provided through an additional library that only works in a server environment. In this regard, it was decided to split the module into two different builds.

The configuration file webpack.conf was created, which describes how to build the project. Below is the code listing for this file.

```
const TerserPlugin = require("terser-webpack-plugin");
const path = require("path");
const { merge } = require("webpack-merge");
const baseConfig = {
  optimization: {
    minimize: true,
```

```

    minimizer: [new TerserPlugin({ extractComments: false })],
  },
  module: {
    rules: [
      {
        test: /\.m?js$/,
        exclude: /node_modules/,
        type: "javascript/auto",
        resolve: {
          fullySpecified: false,
        },
        use: {
          loader: "babel-loader",
        },
      },
    ],
  },
  output: {
    path: path.resolve(__dirname, "dist"),
    library: "ManyCamSDK",
    libraryTarget: "umd",
  },
};

const nodeConfig = merge(baseConfig, {
  target: "node",
  entry: "./index.js",
  output: {
    filename: "ManyCamNode.js",
  },
});

const browserConfig = merge(baseConfig, {
  target: "web",
  entry: "./index.js",
  output: {
    filename: "ManyCam.js",
  },
});

module.exports = [browserConfig];

```

Listing 1 – Webpack.conf file

It contains configuration objects that contain the following fields:

1. Optimization – the output file will be minified to reduce weight.
2. Module – building rules are described here. In this case, the building is done using a transpiler that rewrites the code of the modern Javascript standard (ES2015) to a later one.
3. nodeConfig and browserConfig – the entry point is index.js and the exit point is filename. Two different assemblies are used for the server and client parts.

### 3.4. Project structure

The structure of the project is shown in Figure 11.

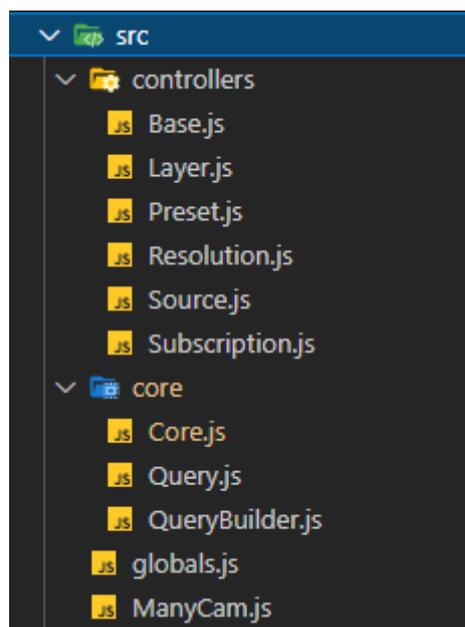


Figure 11 – Source structure

The core directory contains the main logic of the module. The src/core/Core.js file contains the main algorithm for establishing a connection with the API server, handling WebSocket protocol events. There are following events:

1. onerror – occurrence of a connection error.
2. onclose – closing the connection.
3. onopen – opening a connection.

4. onmessage – receiving a message from the API server.

The first three events are used to notify the user of the state of the connection.

The fourth event processes the server response.

The API server accepts a specific message format. It should contain:

1. Action type.
2. Callback function for processing the server response.
3. Access key (optional, for a connection establishing).
4. Transferred data.

To create a request, the data above should be specified. To facilitate this process, it was decided to use the Builder design pattern. Builder is a generative programming pattern. It is used to create complex objects using a chain of methods:

```
someBuilder->setValueA(1)->setValueB(2)->build()
```

Listing 2 – Builder pattern chainig

This pattern was implemented in src/core/QueryBuilder.js file.

There is no need to remember parameters order and which type and structure the data should have. Also, this approach allows you to create objects with only those parameters that are needed for a particular request. This will allow the library to easily scale as new requirements are added.

The src/controllers directory contains classes that represent the main entities in the ManyCam program:

1. Preset – working scene.
2. Layer – preset layer.
3. Source – video source.
4. Subscription – user subscription.
5. Resolution – video resolution.

These entities are needed for manipulations with presets and layers.

The src/globals.js file contains the data references used:

1. Names of video sources.
2. Names of possible colors for setting the background as a video source.

3. Names of transitions between presets.
4. Types of presets.
5. Types of commands to the API server, etc.

The index.js file is used to create bundle files with webpack. The listing is below:

```
const ManyCam = require("./src/ManyCam");
const Preset = require("./src/controllers/Preset");
const Layer = require("./src/controllers/Layer");
const Source = require("./src/controllers/Source");
const Subscription = require("./src/controllers/Subscription");
const Resolution = require("./src/controllers/Resolution");
const {
  sources,
  colors,
  transitionTypes,
  presetTypes,
} = require("./src/globals");

module.exports = {
  ManyCam,
  Preset,
  Source,
  Subscription,
  Layer,
  Resolution,
  sources,
  colors,
  transitionTypes,
  presetTypes,
};
```

Listing 3 – Index.js file

It imports all the modules written during the development process.

1. ManyCam is the main module that implements the main class for working with the API.

2. Preset, Layer, Source, Subscription, Resolution, sources, colors, transitionTypes, presetTypes – auxiliary functions and objects for using the SDK.

### 3.5. Main methods

The src/ManyCam.js module is a class that contains methods that implement all the declared functions of the ManyCam application API.

There are main ones:

1. connect() – connect to the WebSocket protocol;
2. login(account, password) – log in.
3. logout() – deauthorize.
4. exitApp() – exit the application.
5. showUI() – show the application.
6. hideUI() – hide the application in the tray.
7. createPresetInstance() – create a preset object.
8. createLayerInstance() – create a layer object.
9. getAvailableWebcams() – get a list of available cameras.
10. getApplicationSources() – get a list of available resources.
11. getUiState() – get the state of the application (hidden / open).
12. getPresetState(preset) – get preset state.
13. getVideoResolutions() – get a list of available video resolutions.
14. getCurrentResolution() – get current video resolution.
15. setVideoResolution(resolution) – set resolution.
16. getCurrentPreset() – get the current preset.
17. createPreset(preset) – create a preset from an object.
18. editPreset(preset) – edit preset.
19. deletePreset(preset) – delete preset.
20. clearPreset(preset) – clear preset.
21. makePresetActive(preset) – make the preset active.
22. saveSnapshot() – create a screenshot.

- 23.createLayer(layer) – create a layer from an object.
- 24.editLayer(layer) – edit layer.
- 25.deleteLayer(layer) – delete layer.
- 26.onMotionDetected(callback) – an event triggered if motion is detected on the webcam.
- 27.onPresetChanged(callback) – event triggered if the preset has been edited.
- 28.onError(callback) – an event that fires if an error occurs.

Below is a listing of the Connect() method:

```
connect() {
  return new Promise((resolve, reject) => {
    const md5Hash = md5(this.settings.access_key);
    let query = new QueryBuilder(commands.request_for_access)
      .setAccessKey(md5Hash)
      .build();

    if (this.opened()) {
      this.query(query).then((data) => {
        this.status = readyState.connected;
        resolve(data);
      });
    } else {
      this.on(events.opened, () => {
        this.query(query).then((data) => {
          this.status = readyState.connected;
          resolve(data);
        });
      });
    }
  });
}
```

Listing 4 – Connect method

This asynchronous function calculates an md5 hash based on the connection settings. The next step is to connect to the server. If the handshake is successful (the

hash sum is correct, no connection issues), then the WebSocket connection is established and other SDK methods can be executed.

### 3.6. Unit testing

Unit testing is a process in programming that allows you to check the correctness of individual modules of the source code of the program [31].

The idea is to write tests for each nontrivial function or method. This allows to quickly check if another change of the code led to regressions. In other words, if there is a new error in tested function it will be detected by unit testing. This approach facilitates detection and elimination of such errors.

Testing is done using the Jest library [15].

Below is the listing of the function for testing the connection to the WebSocket protocol.

```
describe("connect", () => {
  test("should return true - successful connection", async () => {
    let ManyCam = new ManyCam(settings);
    await expect(ManyCam.connect()).resolves.toBeTruthy();
    ManyCam.close();
  });
});

describe("connected state", () => {
  test("should check if ManyCam state is connected", async () => {
    let ManyCam = new ManyCam(settings);
    await ManyCam.connect();
    expect(ManyCam.connected()).toBeTruthy();
    ManyCam.close();
  });
});
```

Listing 5 – Connection testing functions

The describe method describes which element is being tested.

The test method describes the test itself and implements it.

The expect method expects execution of a certain function to return a certain result. If the result matches expectations, the test is passed.

After test is written, the jest command is run in the console and the test is executed (Figure 12).

```
E:\Study\FRONTEND\TomskSoft\manycam-sdk-js_HUB>node "e:/Study/FRONTEND/TomskSoft/manycam-sdk-js_HUB/node_modules/jest/bin/jest.js" "e:/Study/FRONTEND/TomskSoft/manycam-sdk-js_HUB/test/core/connection.test.js" -t "connect"
PASS test/core/connection.test.js (11.987 s)
  connect
    ✓ should return true - successful connection (156 ms)
  connected state
    ✓ should check if manycam state is connected (82 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        20.934 s
Ran all test suites matching /e:\\Study\\FRONTEND\\TomskSoft\\manycam-sdk-js_HUB\\test\\core\\connection.test.js/i with tests matching "connect".
```

Figure 12 – Testing example

In this case both tests passed successfully.

The tests covered all external SDK functions. The result of all tests is shown in Figure 13.

```

    ✓ should throw on invalid time param (1 ms)
  setVideoResolution
    ✓ should return true on valid resolutions (124 ms)
  autoLaunchGame
    ✓ should return true on valid param (12 ms)

  ● ManyCam > Preset management > saveLastImageOnMotionDetected > should return true on correct param

    expect(received).resolves.toBeTruthy()

    Received promise rejected instead of resolved
    Rejected to value: [Error: Command "save_motion_detected_image" failed: there is no detected image]

  Test Suites: 1 failed, 2 passed, 3 total
  Tests:       1 failed, 60 passed, 61 total
  Snapshots:  0 total
  Time:       17.895 s

```

Figure 13 – All tests

In this case, one test failed, because its performance depends on the conditions of use of ManyCam, and it must be tested separately.

Testing coverage is shown in Figure 14.

83.23% Statements 263/316   60.92% Branches 53/87   74.07% Functions 80/108   84.69% Lines 260/307

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

File	Statements	Branches	Functions	Lines
manycam-sdk-js_HUB	100%	9/9	100%	0/0
manycam-sdk-js_HUB/src	79.7%	106/133	52.5%	21/40
manycam-sdk-js_HUB/src/controllers	82.35%	70/85	63.64%	14/22
manycam-sdk-js_HUB/src/core	87.64%	78/89	72%	18/25

Figure 14 – Testing coverage

It shows that about 60% of functions were covered by unit testing. It will help when SDK gets bigger.

### 3.7. Documentation

Documentation is an essential part of project success. Documentation helps developers use the software.

A special library DocJS [28] is used to document a project.

The documentation of a project is done by commenting the code in a certain way. An example of comments for documenting a class and a function is given in the code listing below:

```
/**
 * Main class containing control methods
 */
class ManyCam extends Core {
  /**
   * @param {Object} settings - Settings
   * @param {string} settings.host - Host
   * @param {string} settings.port - Port
   * @param {string} settings.access_key - Access Key provided by
ManyCam
   */
  constructor(settings) {
    super(settings);
  }
  /**
   * Connects to ManyCam API
   * @returns {Promise<boolean>} Success of the command
   */
  connect() { ... }
```

Listing 6 – Comments for documenting

Comments have a special format with the use of / and \* symbols. The first line is a description, then come following tags:

1. The @param tag allows you to specify the name (mandatory), type and description of the function parameter. The syntax is as follows: @param {Type} Name – Description. The type of the parameter must be a built-in JavaScript type.

2. The @returns tag describes the value returned by the function. Syntax: @returns {Type} Description.

3. The @property tag describes the value of the object. Syntax: @ property {Name} Description.

To build documentation, the `npm run docs` command executes the `createDocs` file. It describes the procedure for creating documentation, templates for generating files, and generation rules. The documentation generation process reads comments from all project files and forms a structured document (Figure 15) with a `.md` extension describing all available functions.

## Classes

---

Name	Description
<a href="#">Layer</a>	Represents a layer in a preset
<a href="#">ManyCam</a>	Main class containing control methods
<a href="#">Preset</a>	Represents preset
<a href="#">Resolution</a>	Represents resolution
<a href="#">Source</a>	Represents source
<a href="#">Subscription</a>	Represents subscription

## Constants

---

Name	Description
<a href="#">colors</a>	Colors
<a href="#">presetTypes</a>	Types of preset
<a href="#">sources</a>	Types of sources
<a href="#">transitionTypes</a>	Types of transitions

## Type definitions

---

Name	Description
<a href="#">notificationCallback</a>	The callback that handles events in ManyCam

Figure 15 – Documentation

Links to the rest of the documented items are also automatically generated.

After commenting the code, the main readme.md file was written with a description of the SDK and useful links to the documentation, the developer's website, and examples (Figure 16).

# About

## Quicklinks

- [ManyCam](#)
- [Documentation](#)
- [Examples](#)

## ManyCam

ManyCam is one of the best solutions to take your remote lectures and broadcasts on streaming sites to the next level, as well as expand the capabilities of video conferencing applications. You can connect multiple cameras and other video sources such as mobile devices and presentations, use virtual backgrounds, add new video layers and scenes, broadcast your desktop, and more.

### [ManyCam](#)

In order to run API server, you need to switch toggle "Start API server" in ManyCam settings:

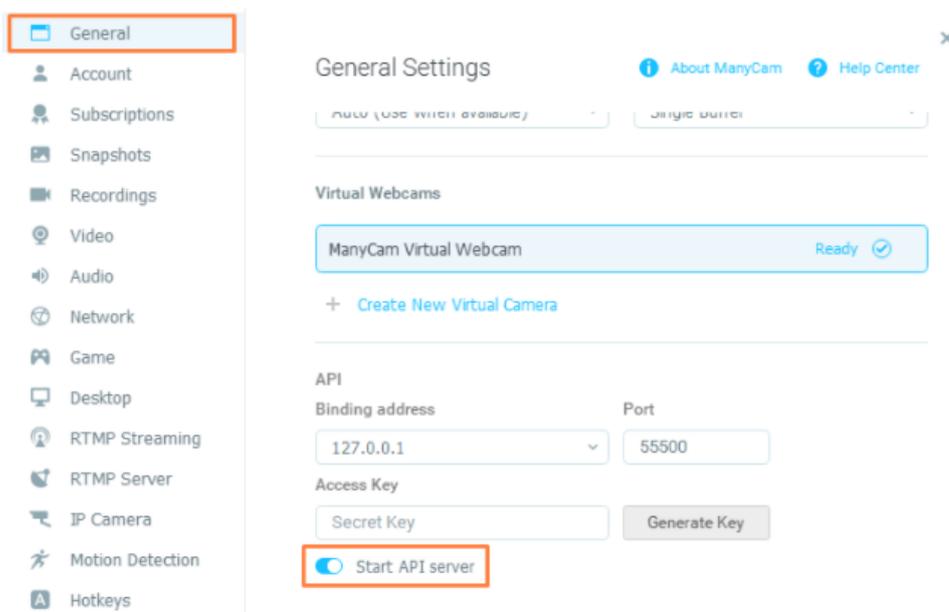


Figure 16 – Readme.md file

### 3.8. Web publication

This SDK is licensed under the MIT License. This is an open source and free software license developed by the Massachusetts Institute of Technology. The MIT license is one of the earliest freely distributable licenses, which is a permissive license, meaning that it allows programmers to use the licensed code in closed source software, provided that the license text is provided with the software.

The SDK has been published on GitHub <https://github.com/tomsksoft-llc/ManyCam-sdk-js> and the NPM package manager <https://www.npmjs.com/package/ManyCam-sdk>.

## 4. SDK usage for development

### 4.1. SDK setup

To use applications on Node.js, it is required to install the package from npm:

```
npm install ManyCam-sdk
```

Listing 7 – Install SDK

Then import it in file:

```
const ManyCamSDK = require("ManyCam-sdk");  
//or using destruction:  
const { ManyCam } = require("ManyCam-sdk");
```

Listing 8 – Import SDK

If development is carried out on the client side (for the browser), it is important to use other option – including the following code in the HTML markup:

```
<script type="text/javascript" src="dist/ManyCam.js"></script>  
//Or from CDN:  
<script  
  type="text/javascript"  
  src="https://cdn.jsdelivr.net/npm/ManyCamsdk@latest/dist/ManyCam.js"  
></script>  
  
// The ManycamSDK can then be initiated like this:  
<script>  
  const ManyCam = new ManyCamSDK.ManyCam(settings);  
  // ...  
</script>
```

Listing 9 – Browser use

### 4.2. Demo Application

Using the Vue.js Framework, an application was created to demonstrate the capabilities of the SDK (Figure 17).

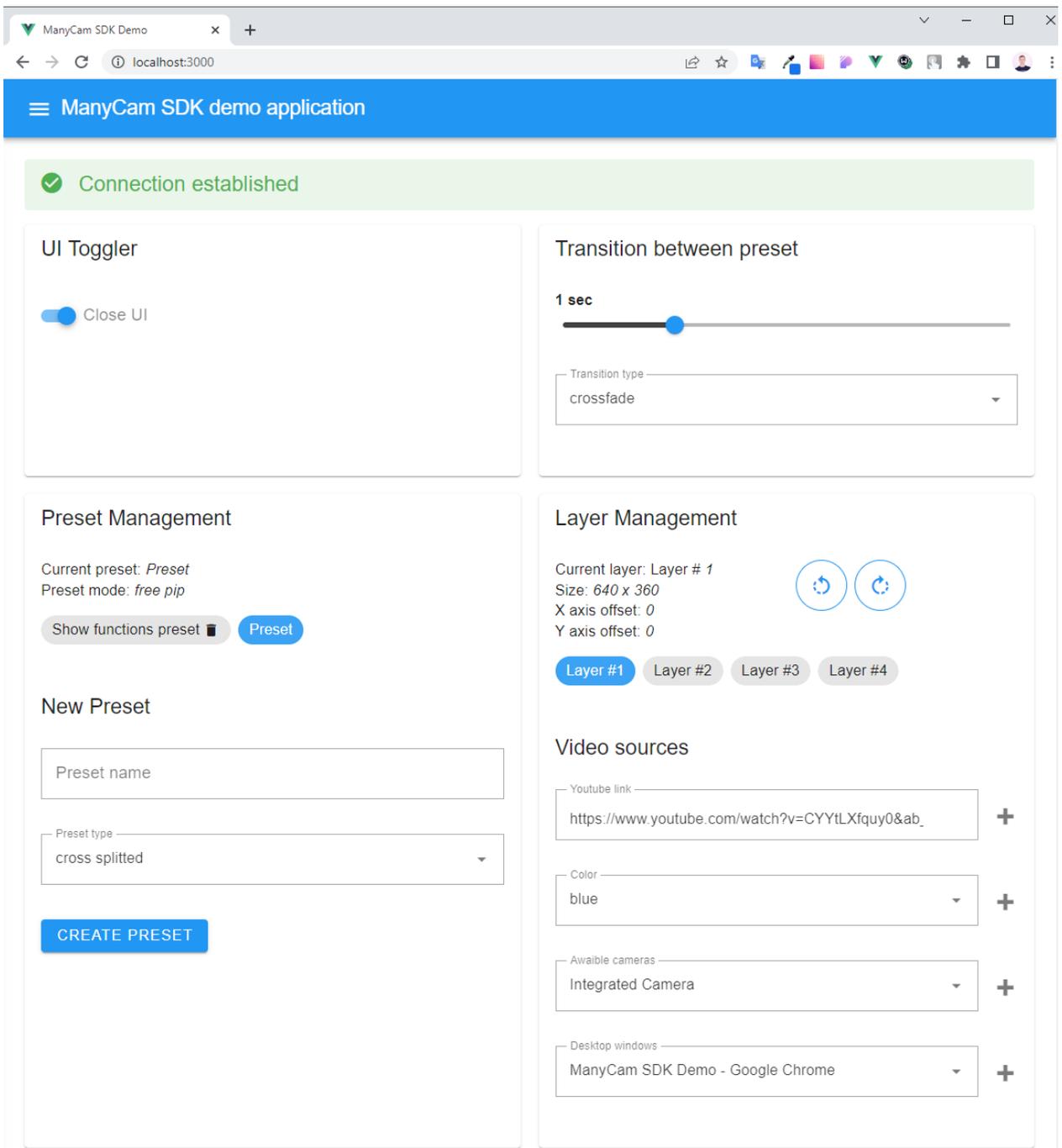


Figure 17 – Demo Vue.js app

In this application, 4 semantic blocks are implemented from left to right:

1. Switching window modes of the ManyCam application.
2. Setting the duration and selection of animation for transitions between presets.

3. Create and delete presets. It is possible to switch between presets by clicking on the label with name of preset. Deletion is done by clicking on the trash icon next to the name.
4. Manage layers in a preset: general information about a layer, layer selection, video source setting. The video source can be YouTube video, static background fill, video camera image and screen sharing. It is possible also to rotate the layer by 90 degrees.

The preset management block allows to create and edit a preset and its type. For demonstration, a preset was created with the name "New preset" and the type "horizontally splitted" (Figure 18). This means that the created preset has two layers separated horizontally.

## Preset Management

Current preset: *New preset*

Preset mode: *horizontally splitted*



### New Preset

Preset name

Preset type



Figure 18 – Preset management

Next, in the layer control block, video sources were assigned to each layer: web-cam and YouTube video. The video was then rotated by 180 degrees (Figure 19)

# Layer Management

Current layer: Layer # 1  
Size: 1280 x 360  
X axis offset: 0  
Y axis offset: 0



Layer #1    Layer #2

## Video sources

Youtube link  +

Color  ▼ +

Available cameras  ▼ +

Desktop windows  ▼ +

Figure 19 – Edit layers

The result of these actions is shown in Figure 20.

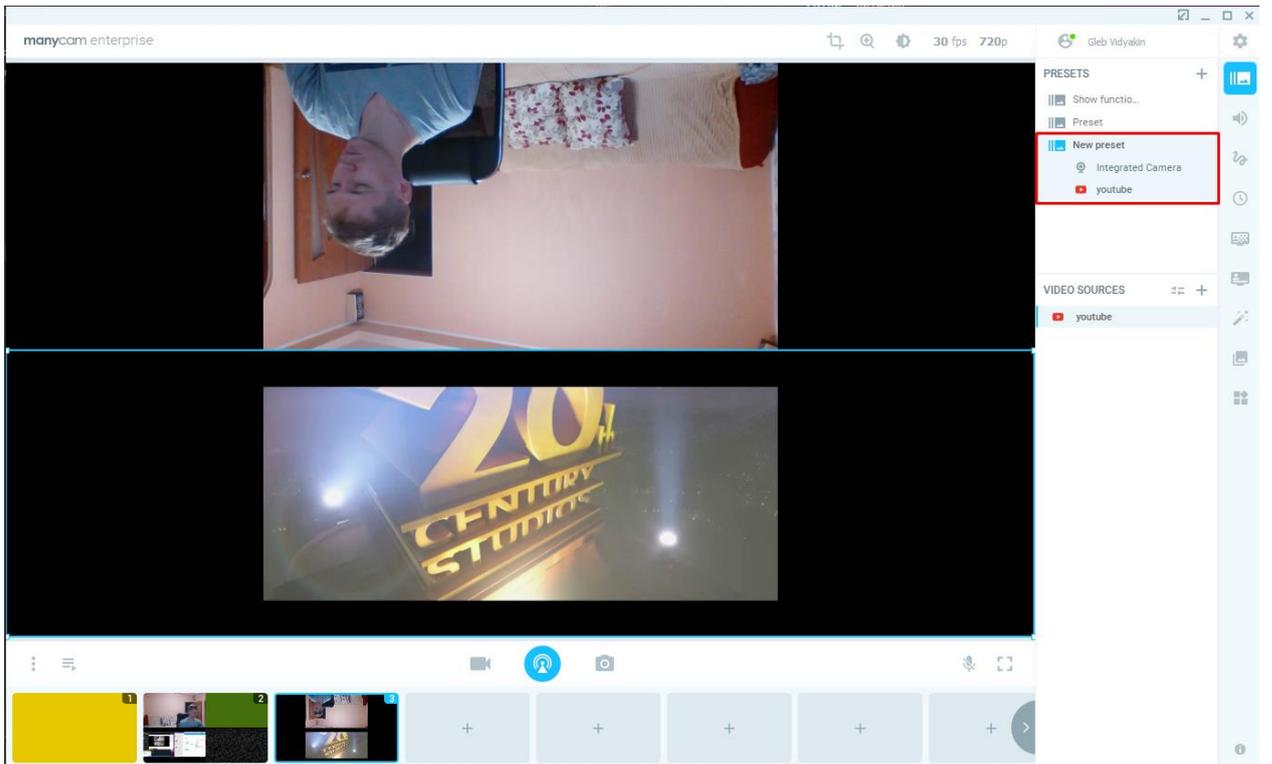


Figure 20 – Result

## **5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение**

### **Введение**

В настоящее время стриминговые приложения довольно актуальны. Для привлечения большего числа клиентов необходима пользовательская интеграция продукта.

Цель раздела данного раздела – оценка эффективности, рисков, экономической успешности проекта.

Чтобы достичь поставленной цели необходимо решить задачи по организации работы над проектным решением, по планированию этапов разработки, оценить перспективность и коммерческий потенциал проекта, рассчитать бюджет, необходимый для реализации проекта, оценить социальную и экономическую эффективность проекта.

Цель данной НИР спроектировать и разработать модуль для веб-интеграции приложения «ManyCam». Данный модуль будет использован для интеграции приложения в пользовательские решения без привлечения разработчиков со стороны команды «ManyCam».

### **5.1. Оценка коммерческого и инновационного потенциала НИИ**

Перед планированием работы, определением ресурсного и экономического потенциала разработки программно-алгоритмического комплекса, следует уделить особое внимание оценки коммерческого потенциала и перспективности новой разработки в целом, дать характеристику и определить сегмент рынка, на который будет ориентироваться компания, при продаже данной продукции.

### 5.1.1. Потенциальные потребители результатов исследования

Целевой рынок можно разделить на три сегмента, по характеру применения разработки:

1. Научные и образовательные учреждения – некоммерческие научно-практические разработки;
2. Компании разработчики ПО – коммерческие научно-практические разработки;
3. Частные пользователи – практическое применение;

Карта сегментирования рынка следующая (т 1).

Таблица 1 – Карта сегментирования рынка продаж

Использование продукции Группа потребителей	Некоммерческая разработка	Коммерческая разработка	Практическое использование
ВУЗы и НИИ			
Разработчики ПО			
Частные пользователи			

ТПУ	Cisco	Репетитор

Согласно карте сегментирования рынка, был сделан вывод об относительной свободе сегмента использования модуля интеграции приложения.

### 5.1.2. Анализ конкурентных технических решений

Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения основывается на сравнении разрабатываемого модуля (ф) и двух конкурентных решений стриминговых приложений, а именно «MUX» (к1) и «Аgоgа» (к2).

Таблица 2 – Анализ конкурентных технических решений

Критерий оценки	Вес критерия	Баллы			Конкурентно-способность		
		Б <sub>ф</sub>	Б <sub>к1</sub>	Б <sub>к2</sub>	К <sub>ф</sub>	К <sub>к1</sub>	К <sub>к2</sub>
<b>Технические критерии оценки ресурсоэффективности</b>							
1. Использование постоянного соединения	0,1	5	1	1	0,5	0,1	0,1
2. Использование при повышенной нагрузке	0,1	2	3	4	0,2	0,3	0,4
3. Ресурсоэффективность	0,1	3	4	5	0,3	0,4	0,5
4. Простота эксплуатации	0,1	5	4	4	0,5	0,4	0,4
5. Масштабируемость	0,05	6	4	5	0,3	0,2	0,25
6. Многопрофильность	0,1	3	5	5	0,1	0,2	0,3
<b>Экономические критерии оценки эффективности</b>							
1. Конкурентоспособность продукта	0,1	3	5	5	0,3	0,5	0,5
2. Уровень проникновения на рынок	0,05	2	4	5	0,1	0,2	0,25
3. Цена	0,1	5	3	4	0,5	0,3	0,4
4. Предполагаемый срок эксплуатации	0,1	5	5	5	0,5	0,5	0,5
5. Послепродажное обслуживание	0,1	3	5	5	0,3	0,55	0,55
<b>Итого</b>	<b>1</b>	<b>42</b>	<b>43</b>	<b>48</b>	<b>3,6</b>	<b>3,7</b>	<b>4,2</b>

Разработка уступает индустриальным конкурентам, но разрыв позволяет предположить о возможности продукта конкурировать с ними. Кроме того, по отдельным пунктам: масштабируемость, потребление вычислительных мощностей и цена, разработка опережает конкурентов.

### 5.1.3. SWOT анализ

Проведём исследование внутренних и внешних свойств проекта с помощью методики SWOT-анализа. Матрица SWOT-анализа описывает сильные и слабые стороны проекта (внутренние факторы), а также показывает

возможности и угрозы (внешние факторы) и возможные направления реализации.

В рамках первого этапа анализа была построена матрица SWOT ( 3) с описанием сильных и слабых сторон проекта, а также возможностей и угроз.

Таблица 3 – Матрица SWOT

	<b>Сильные стороны:</b> С1. Использование постоянного соединения; С2. Возможность внедрения в другие программные комплексы; С3. Расширяемость системы; С4. Спонсирование по грантовой программе.	<b>Слабые стороны:</b> Сл1. Опыт в разработке. Сл2. Использование зарубежного ПО в разработке. Сл3. Зависимость от API приложения
<b>Возможности:</b> В1. Заинтересованность крупных предприятий Выход на международный рынок В2. Выход на международный рынок В3. Сотрудничество с российскими разработчиками В4. Возможность расширить SDK		
<b>Угрозы:</b> У1. Наличие сильных конкурентов; У2. Ограничения на экспорт разработки; У3. Введение дополнительных государственных требований к сертификации ПО У4. Исчерпание финансирования.		

В рамках второго этапа был проведен анализ соответствия сильных и слабых сторон проекта внешним условиям окружающей среды. Анализ

представлен ниже в виде интерактивных матриц, сильное соответствие отмечено знаком «+», слабое знаком «-».

Таблица 4 – Интерактивная матрица сильных сторон и возможностей проекта

Сильные стороны					
Возможности		C1	C2	C3	C4
	B1	+	-	+	+
	B2	+	+	+	+
	B3	+	-	+	+
	B4	-	+	+	-

Направления реализации сильных сторон и возможностей: B1B2B3C1C3C4, B2B4C2C3C4.

Таблица 5 – Интерактивная матрица слабых сторон и возможностей проекта

Сильные стороны					
Возможности		C1	C2	C3	C4
	B1	+	-	+	+
	B2	+	+	+	+
	B3	+	-	+	+
	B4	-	+	+	-

Направления реализации слабых сторон и возможностей: B2B4Cл1, B1B2Cл2, B2B4Cл3.

Таблица 6 – Интерактивная матрица сильных сторон и угроз проекта

Сильные стороны					
Угрозы		C1	C2	C3	C4
	У1	+	+	+	-
	У2	-	-	+	+
	У3	-	-	+	+
	У4	+	+	+	-

Направления реализации сильных сторон и угроз: У1C1C2C3, У2У3C3C4, У4C1C2C3.

Таблица 7 – Интерактивная матрица слабых сторон и угроз проекта

Слабые стороны				
Угрозы		Сл1	Сл2	Сл3
	У1	+	+	-
	У2	-	-	+
	У3	-	-	-
	У4	+	+	+

Направления реализации слабых сторон и угроз: У1У4Сл1Сл2, У2У4Сл3.

В рамках третьего этапа была построена итоговая матрица SWOT анализа, результат представлен в таблице (т 8)

Таблица 8 – Итоговая матрица SWOT

	<b>Сильные стороны:</b> С1. Использование постоянного соединения; С2. Возможность внедрения в другие программные комплексы; С3. Расширяемость системы; С4. Спонсирование по грантовой программе.	<b>Слабые стороны:</b> Сл1. Опыт в разработке. Сл2. Использование зарубежного ПО в разработке. Сл3. Зависимость от API приложения
<b>Возможности:</b> В1. Заинтересованность крупных предприятий Выход на международный рынок В2. Выход на международный рынок В3. Сотрудничество с российскими разработчиками В4. Возможность расширить SDK	В1В2В3С1С3С4 – Внедрение ПО как в России так и за рубежом. В2В4С2С3С4 – Дальнейшие разработки, расширение функционала, интеграция с другими продуктами.	В2В4Сл1 – Привлечение специалистов, совместные разработки с другими компаниями; В1В2Сл2 – Рекламная кампания успешных внедрений; В2В4Сл3 – Переход на отечественное ПО по мере развития проекта.
<b>Угрозы:</b> У1. Наличие сильных конкурентов; У2. Ограничения на экспорт разработки; У3. Введение дополнительных государственных требований к сертификации ПО У4. Исчерпание финансирования.	У1С1С2С3 – наращивание функционала, интеграция; У2У3С3С4 – Ориентирование разработки на внутренний рынок; У4С1С2С3 – лицензирование, продажа прав на использование интеллектуальной собственности.	У1У4Сл1Сл2 – Сотрудничество с разработчиками ПО планирования и моделирования, обмен опытом; У2У4Сл3 – Привлечение финансирования из гос. сектора на импортозамещение.

#### 5.1.4. Оценка готовности проекта к коммерциализации

Для оценки готовности проекта были определены показатели по вопросам в таблице 9. Оценка проводится по пятибалльной шкале. При оценке научного проекта: 1 балл – не проработано, 2 балла – проработка слабая, 3 балла – выполнено, качество посредственное, 4 балла – удовлетворительное качество, 5 баллов – качество подтверждено сторонним специалистом. При

оценке знаний разработчика: 1 балл – не знаю, 2 балла – только теоретические знания, 3 балла – теоретические знания с практическими примерами, 4 балла – умею, практикую, 5 баллов – могу консультировать по вопросу.

Таблица 9 – Таблица оценки готовности научного проекта к коммерциализации

№ п/п	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
1	Определён имеющийся научно-технический задел	4	5
2	Определены перспективные направления коммерциализации научно-технического задела	3	3
3	Определены отрасли и технологии (товары, услуги) для представления на рынок	3	3
4	Определена товарная форма научно-технического задела для представления на рынок	2	2
5	Определены авторы и осуществлена охрана их прав	4	3
6	Проведена оценка стоимости интеллектуальной собственности	3	3
7	Проведены маркетинговые исследования рынков сбыта	3	3
8	Разработан бизнес-план коммерциализации научной разработки	4	3
9	Определены пути продвижения научной разработки на рынок	2	2
10	Разработана стратегия реализации научной разработки	4	5
11	Проработаны вопросы международного сотрудничества и выхода на зарубежный рынок	2	2
12	Проработаны вопросы использования инфраструктуры поддержки, получения льгот	2	1
13	Проработаны вопросы финансирования коммерциализации научной разработки	3	3
14	Имеется команда для коммерциализации научной разработки	5	5

15	Проработан механизм реализации научной разработки	4	4
<b>ИТОГО</b>		48	47

Итоговые результаты показывают проработанность проекта выше среднего. Для успешного дальнейшего продвижения необходимо развить компетенции в области экономики и маркетинга.

## 5.2. Инициация проекта

В данном этапе фиксируются начальные цели, содержание и финансовые ресурсы. Определяются заинтересованные стороны, которые могут повлиять на конечный результат проекта. Эта информация закрепляется в уставе проекта.

### 5.2.1. Цели и результат проекта

Были определены заинтересованные стороны ( 10). Заинтересованные стороны – это лица или организации, которые активно заинтересованы и/или могут быть как положительно, так и отрицательно затронуты в результате проекта.

Таблица 10 – Заинтересованные в проекте стороны

<b>Заинтересованные стороны</b>	<b>Ожидания</b>
Томский политехнический университет	Лицензирование научных разработок, публикация научных материалов.
ООО “ТомскСофт”	Программный пакет для практического применения

В таблице 11 представим цель и результаты проекта, а также критерии их достижения и требования к результатам.

Таблица 11 – Цели и результаты проекта

Цель проекта:	Проектирование и разработка модуль для веб-интеграции приложения «ManyCam
Ожидаемые результаты:	Программный продукт, обеспечивающий возможность интеграции приложения «ManyCam» на языке JS как на клиентских приложениях, так и серверных.

Критерии приёмки результатов:	1. Уменьшение времени интеграции. 2. Понижение затрат на частную интеграцию приложения.
Требования к результатам:	1. Разработать клиентскую часть модуля предпросмотра. 2. Разработать модуль рендеринга изображений для обработки на сервере. 3. Разработать демонстрационный вариант внедренного приложения

### 5.2.2. Организационная структура проекта

В таблице 12 отражена организационная структура, роль и функции каждого члена команды.

Таблица 12 – Рабочая группа

№ п/п	ФИО, основное место работы, должность	Роль	Функции	Трудовые затраты, час.
1	Кочегурова Елена Алексеевна, Томский политехнический университет, доцент	Руководитель	Заверение документов, определение направления развития проекта.	36
2	Видякин Глеб Игоревич, Томский политехнический университет, магистр	Исполнитель	Разработка ПО, документирование результатов.	540

### 5.2.3. Ограничения и допущения

Определи факторы, которые могут послужить ограничением степени свободы участников команды проекта ( 12).

Таблица 13 – Ограничения проекта

Фактор	Ограничения / допущения
--------	-------------------------

Бюджет	250 т.р.
Источники	ООО «ТомскСофт»
Сроки	27.12.2019 – 04.06.21
Дата утверждения плана управления проектом	01.02.21
Дата завершения	04.06.22
Прочие ограничения	Использование технологии Node.js в реализации модуля; Реализация SDK в виде подключаемой библиотеки

### 5.3. Планирование управления НТИ

Для планирования хода работ НТИ определена структура работ, распланированы контрольные события и определен календарный план проекта, отраженный в диаграмме Ганта.

#### 5.3.1. Контрольные события проекта

Определены ключевые события проекта, их даты и результаты в таблице 14.

Таблица 14 – Контрольные события проекта

№ п/п	Контрольное событие	Дата	Результат
1	Сбор данных о предметной области	01.02.21	Раздел ВКР
2	Создание основного модуля	01.03.21	Раздел ВКР
3	Создание тестов	01.07.21	Раздел ВКР
4	Создание документации	01.09.21	Раздел ВКР
5	Создание демонстрационного приложения	01.12.21	Раздел ВКР
6	Формирование отчёта по разработке	04.03.22	ВКР

### **5.3.2. План проекта**

В виде диаграммы Ганта был составлен линейный график работ по проекту, в котором отражены даты начала и окончания, длительность и ответственных лиц по каждому этапу работ (рис.1).



### 5.3.3. Бюджет НИИ

Бюджет научного исследования должен в полной мере отражать все планируемые расходы на его выполнение. Бюджет формируется по следующим статьям: сырьё и материалы, специальное оборудование, основная заработная плата, дополнительная заработная плата, отчисления в социальные фонды, командировки, оплата работ сторонних организаций.

#### Сырьё и материалы

Разработка программного обеспечения проводится на персональном компьютере без использования материального сырья. Расчёт данной статьи расходов не требуется.

#### Специальное оборудование

В качестве специального оборудования подразумеваются лицензии на ПО и затраты на приобретение ПК для разработки.

№ п/п	Наименование	Кол-во, шт.	Цена единицы, руб.	Общая стоимость, руб.
1	Лицензия GitKraken на месяц	12	300,00	3600,00
2	ПК	1	80000,00	60000,00
<b>Итого:</b>				<b>63600.00</b>

#### Основная заработная плата

Заработная плата работника включает основную заработную плату, и дополнительную заработную плату:

$$Z_{\text{зп}} = Z_{\text{осн}} + Z_{\text{доп}} \quad (1.1)$$

Где  $Z_{\text{осн}}$  – основная заработная плата;

$Z_{\text{доп}}$  – дополнительная заработная плата (12-20 % от  $Z_{\text{осн}}$ ).

Основная заработная плата ( $Z_{\text{осн}}$ ) руководителя (лаборанта, инженера) от предприятия (при наличии руководителя от предприятия) рассчитывается по следующей формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} \cdot T_p \quad (1.2)$$

Где  $Z_{\text{осн}}$  – основная заработная плата одного работника, руб.;

$T_p$  – продолжительность работ, выполняемых научно-техническим работником, раб. дн. (таблица);

$Z_{\text{дн}}$  – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_m \cdot M}{F_d} \quad (1.3)$$

Где  $Z_m$  – месячный должностной оклад работника, руб.;

$M$  – количество месяцев работы без отпуска в течение года: при отпуске в 48 раб. дня  $M = 10,4$  месяца, 6-дневная неделя;

$F_d$  – действительный годовой фонд рабочего времени научно-технического персонала, рабочие дни (таблица 15).

Таблица 15 – Баланс рабочего времени

Показатели рабочего времени	Руководитель	Студент
Календарное число дней	365	365
Количество нерабочих дней		
- выходные дни	118	118
- праздничные дни		
Потери рабочего времени		
- отпуск	48	48
- невыходы по болезни		
Действительный годовой фонд рабочего времени	199	199

Месячный должностной оклад работника:

$$Z_m = Z_{\text{тс}} \cdot (1 + k_{\text{пр}} + k_d) \cdot k_p \quad (1.4)$$

Где  $Z_{\text{тс}}$  – заработная плата по тарифной ставке, руб.;

$k_{\text{пр}}$  – премиальный коэффициент, равный 0,3 (т.е. 30% от  $Z_{\text{тс}}$ );

$k_d$  – коэффициент доплат и надбавок составляет примерно 0,2 – 0,5 (в НИИ и на промышленных предприятиях – за расширение сфер обслуживания, за профессиональное мастерство, за вредные условия: 15-20 % от  $Z_{\text{тс}}$ );

$k_p$  – районный коэффициент, равный 1,3 (для Томска).

Для предприятий, не относящихся к бюджетной сфере, тарифная заработная плата (оклад) рассчитывается по тарифной сетке, принятой на данном предприятии. Расчёт основной заработной платы приведён в таблице 16.

Таблица 16 – Расчёт основной заработной платы

Исполнители	$Z_{тс}$ , руб.	$k_{пр}$	$k_d$	$k_p$	$Z_m$ , Руб	$Z_{дн}$ , руб.	Тр, раб.	$Z_{осн}$ , руб.
Руководитель	33 664,00	0,3	0,15	1,3	63 456,64	3 316,33	14	46 428,58
Студент	25 000,00	0,3	0,15	1,3	43 750,50	2286,45	60	137187,49
<b>Итого:</b>								183616,07

### Дополнительная заработная плата

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{доп} = k_{доп} \cdot Z_{осн}. \quad (1.5)$$

Где  $k_{доп}$  – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12 – 0,15).

Примем коэффициент равный 0,12. Результаты расчета приведены в таблице 17

Таблица 17 – Расчёт дополнительной заработной платы

Исполнители	$Z_{тс}$ , руб.	$k_{пр}$	$k_d$	$k_p$	$Z_m$ , Руб	$Z_{дн}$ , руб.	Тр, раб.	$Z_{осн}$ , руб.
Руководитель	33 664,00	0,3	0,15	1,3	63 456,64	3 316,33	14	46 428,58
Студент	25 000,00	0,3	0,15	1,3	43 750,50	2286,45	60	137187,49
<b>Итого:</b>								183616,07

### Отчисления во внебюджетные фонды

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$Z_{внеб} = k_{внеб} \cdot (Z_{осн} + Z_{доп}). \quad (1.6)$$

Где  $k_{внеб}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Результаты расчета приведены отчислений и общий итог по фонду заработной платы приведены в таблице 18.

Таблица 18 – Общий итог по зарплатному фонду

<b>Исполнитель</b>	<b>Коэффициент отчислений во внебюджетные фонды</b>	<b>Основная заработная плата, руб.</b>	<b>Дополнительная заработная плата, руб.</b>	<b>Отчисления во внебюджетные фонды</b>
<b>Руководитель проекта</b>	0,3	46 428,58	5 571,43	15 600,00
<b>Студент</b>		137187,49	15 101,98	45686.84
<b>Итог:</b>		183616.07	20 673,41	61286.84

### **Научные и производственные командировки**

Данный вид работ не запланирован, расчёт статьи расходов не требуется.

### **Контрагентные расходы**

Привлечение сторонних организаций не запланировано, расчёт статьи расходов не требуется.

### **Накладные расходы**

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$Z_{накл} = k_{накл} \cdot (Z_{осн} + Z_{доп}). \quad (1.7)$$

Где  $k_{накл}$  – коэффициент, учитывающий накладные расходы.

Примем коэффициент равным 0,8. Таким образом сумма накладных расходов составит (таблица):

Таблица 19 – Расчёт накладных расходов

Исполнитель	Коэффициент учёта накладных расходов	Основная заработная плата, руб.	Дополнительная заработная плата, руб.	Накладные расходы
Руководитель проекта	0,8	46 428,58	5 571,43	41 600,00
Студент		137187,49	15 101,98	121831.57
<b>Итог:</b>		183616.07	20 673,41	163431.57

### Бюджет НТИ

Статьи, рассчитанные в предыдущих пунктах, сведены в таблице 20.

Таблица 20 – Итоговый бюджет НТИ

Статьи							
Сырье и материалы, руб.	Специальное оборудование, руб.	Основная заработная плата, руб.	Дополнительная заработная плата, руб.	Отчисления на социальные нужды, руб.	Научные и производственные, руб.	Контрагентные расходы, руб.	Накладные расходы, руб.
0,00	63600,00	183616.07	20 673,41	61286.84	0,00	0,00	163431.57
<b>Итог:</b>							492607,89

### 5.3.4. Риски проекта

Составлен реестр возможных рисков (таблица 21).

Таблица 21 – Реестр рисков проекта

№	Риск	Потенциальное воздействие	Вероятность наступления (1-5)	Влияние риска (1-5)	Уровень риска*	Способы смягчения риска	Условия наступления
---	------	---------------------------	-------------------------------	---------------------	----------------	-------------------------	---------------------

1	Несоответствие модели реальным процессам	Увеличение сроков разработки	2	5	Средний	Привлечение дополнительных специалистов	Ошибки в проектировании и архитектуры
2	Критические ошибки приложения	Отказ приложения	1	5	Средний	Привлечение дополнительных специалистов	Ошибки в разработке приложения
3	Невозможность доступа к приложению	Невозможность пользования приложением	1	3	Низкий	Нет	Отказ сетей связи
	Срыв сроков разработки	Увеличение сроков разработки	2	2	Средний	Привлечение дополнительных специалистов	Нарушение сроков разработки

#### 5.4. Определение ресурсной, финансовой, экономической эффективности

Сравнительная эффективность разработки выражается в интегральном показателе эффективности. Этот показатель состоит из двух средневзвешенных величин:

Определение интегральных показателей эффективности проведём в сравнении со следующими аналогами:

1. SDK для видео-стриминговой интеграции Mux Video;
2. SDK для видео-стриминговой интеграции Agora.

Интегральный финансовый показатель разработки определяется как:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}} \quad (1.8)$$

Где  $I_{\text{финр}}^{\text{исп.}i}$  – интегральный финансовый показатель разработки;

$\Phi_{pi}$  – стоимость  $i$ -го варианта исполнения;

$\Phi_{\text{max}}$  – максимальная стоимость исполнения научно-исследовательского проекта.

Результаты вычислений приведены в таблице 22.

Таблица 22 – Расчёт интегрального финансового показателя

Продукт	$\Phi_{pi}$	$\Phi_{max}$	$I_{финр. i}^{исп. i}$
НТИ	430 000,00	1 560 000,00	0,28
Mux Video	1 560 000,00		1,00
Agora	1 00 000,00		0,64

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum a_i \cdot b_i. \quad (1.9)$$

Где  $I_{pi}$  – интегральный показатель ресурсоэффективности для  $i$ -го варианта исполнения разработки;  
 $a_i$  – весовой коэффициент  $i$ -го варианта исполнения разработки;  
 $b_i$  – балльная оценка  $i$ -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;  
 $n$  – число параметров сравнения.

Расчёт показателя приведён в таблице 23.

Таблица 23 – Сравнительная оценка характеристик продуктов

Критерии	Весовой коэффициент параметра	НТИ	Mux Video	Agora
1. Надежность алгоритмов	0,3	3	5	5
2. Быстродействие	0,3	4	5	3
3. Удобство	0,1	4	5	5
4. Функциональность	0,2	2	4	5
5. Интерфейс	0,1	5	5	3
<b>ИТОГО</b>	1	3,4	4,8	4,2

Интегральный показатель эффективности вариантов исполнения разработки ( $I_{исп. i}$ ) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп. 1} = \frac{I_{p-исп1}}{I_{финр. i}^{исп. i}}, \quad I_{исп. 1} = \frac{I_{p-исп1}}{I_{финр. i}^{исп. i}} \text{ и т.д.} \quad (1.10)$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта (таблица 24) и выбрать наиболее целесообразный вариант из предложенных.

Таблица 24 – Расчёт интегрального показателя эффективности

	$I_{\text{финр}}^{\text{исп.}i}$	$I_{\text{р-исп}}$	$I_{\text{исп}}$
НТИ	0,28	3,4	12,14
Mux Video	1,00	4,8	4,80
Agora	0,64	4,2	6.52

Сравнительную эффективность проекта определим по следующей формуле:

$$\mathcal{E}_{\text{ср}} = \frac{I_{\text{финр}}^{\text{р}}}{I_{\text{финр}}^{\text{а}}}; \quad (1.11)$$

Где  $\mathcal{E}_{\text{ср}}$  – сравнительная эффективность проекта;

$I_{\text{тэ}}^{\text{р}}$  – интегральный показатель разработки;

$I_{\text{тэ}}^{\text{а}}$  – интегральный технико-экономический показатель аналога.

Таким образом сравнительная эффективность проекта с аналогами:

1. В сравнении с Mux Video  $\mathcal{E}_{\text{ср}} = 2.52$ ;
2. В сравнении с Agora  $\mathcal{E}_{\text{ср}} = 3.83$ ;

## Вывод

В ходе работы были выявлены направления возможного развития проекта для повышения его конкурентоспособности. Разработка данного проекта является целесообразной с точки зрения экономической успешности, эффективности и рисков.

## **6. Социальная ответственность**

### **Введение**

Целью работы является разработка программного обеспечения. Разработка ведется с использованием компьютерной техники. Рабочее место – это офисный кабинет, стол и компьютер. Работая за компьютером, человек подвергается воздействию ряда опасных и вредных производственных факторов: электромагнитных полей, радиочастотному (высоких, ультравысоких и средних частот) и инфракрасному излучению, шуму и вибрации, статическому электричеству. Работа с компьютером связана с напряжением зрительных органов, нагрузкой на кисти рук при работе с периферийными устройствами, нагрузкой на позвоночник при длительном сидении и умственным напряжением.

### **6.1. Правовые и организационные вопросы обеспечения безопасности**

#### **6.1.1. Особенности законодательного регулирования проектных решений**

Режим труда и отдыха для вида трудовой деятельности на ПЭВМ регламентируется ТК РФ [38], СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания [34] и ТОИ Р-45-084-01 Типовая инструкция по охране труда при работе на персональном компьютере [37].

Работа в офисе относится ко второй категории тяжести труда – работы выполняются при оптимальных условиях внешней производственной среды и при оптимальной величине физической, умственной и нервно-эмоциональной нагрузки. Продолжительность рабочего дня работников не должна превышать 40 часов в неделю.

Продолжительность непрерывной работы за компьютером без регламентированного перерыва не должна превышать 1 час.

Рекомендуется делать перерывы в работе за ПК продолжительностью 10-15 минут через каждые 45-60 минут работы.

Во время регламентированных перерывов целесообразно выполнять комплексы упражнений и осуществлять проветривание помещения.

Эффективными являются нерегламентированные перерывы (микропаузы) длительностью 1-3 минуты. Регламентированные перерывы и микропаузы целесообразно использовать для выполнения комплекса упражнений и гимнастики для глаз, пальцев рук, а также массажа. Комплексы упражнений целесообразно менять через 2-3 недели.

Не рекомендуется работать за компьютером более 6 часов за смену. Для того чтобы ПЭВМ соответствовали нормам, осуществляется производственный контроль и надзор внутри предприятия-производителя. Эксплуатирующие предприятия также следят за характеристиками используемой аппаратуры.

### **6.1.2. Организационные мероприятия при компоновке рабочей зоны**

В соответствии с ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования [22] рабочий стол может быть любой конструкции, отвечающей современным требованиям эргономики и позволяющей удобно разместить на рабочей поверхности оборудование с учетом его количества, размеров и характера выполняемой работы. Выполнение требований на данном рабочем месте отражено ниже в таблице Таблица 25, согласно ТООИ Р-45-084-01 Типовая инструкция по охране труда при работе на персональном компьютере [37] и ГОСТ 12.2.032-78 Система

стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования [22].

Таблица 25 – Требования к организации рабочего места при работе с ПЭВМ

<b>Требование</b>	<b>Требуемое значение</b>	<b>Значение параметров в помещении</b>
Высота рабочей поверхности стола	Регулируемая высота (680-800мм) Нерегулируемая высота (725мм)	Соответствует
Рабочий стул	Подъемно-поворотный, регулируемый по высоте и углу наклона спинки	Соответствует
Расположение монитора от глаз пользователя	600-700мм	Соответствует

Данные требования выполняются на текущем рабочем месте.

## **6.2. Производственная безопасность**

Ниже приведен перечень опасных и вредных факторов, характерных для производственной среды в виде таблицы (таблица 26).

Таблица 26 – Опасные и вредные факторы при выполнении работ по разработке программно-алгоритмического комплекса планирования производственных процессов

Факторы (ГОСТ 12.0.003-2015) [16]	Нормативные документы
---	-----------------------

Недостаточная освещённость рабочей зоны; отсутствие или недостаток естественного света	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95 [36]
Повышенный уровень шума	ГОСТ 12.1.003-83 Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности [17]
Повышенный уровень электро-магнитных излучений	ГОСТ 12.1.006-84 «Система стандартов безопасности труда (ССБТ). Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля» [19]
Повышенная или пониженная влажность воздуха	СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания [34]
Статические перегрузки, умственные перегрузки, перегрузки анализаторов;	МР 2.2.9.2311 – 07 «Профилактика стрессового состояния работников при различных видах профессиональной деятельности» [32]
Электрический ток	ГОСТ 12.1.030-81 Система стандартов безопасности труда (ССБТ). Электробезопасность. Защитное заземление. Зануление [20]
Короткое замыкание	ГОСТ 12.1.030-81 Система стандартов безопасности труда (ССБТ). Электробезопасность. Защитное заземление. Зануление [20]
Статическое электричество	ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно

	допустимые значения напряжений прикосновения и токов [21]
--	---

### **6.2.1. Недостаточная освещённость рабочей зоны; отсутствие или недостаток естественного света;**

Рабочее место находится на 5 этаже здания. Естественное освещение в кабинете присутствует. Основное освещение в аудитории производится посредством общего искусственного освещения.

В процессе разработки, разработчик все время работы пользуется ПК. При этом согласно СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95 [36] освещение не создает бликов на поверхности экрана ПЭВМ, а освещенность экрана не более 300 лк. Кроме того, работа за ПЭВМ относится к категории зрительных работ к разряду Б зрительных работ (восприятие информации).

### **6.2.2. Повышенный уровень шума**

Звуковые колебания, издаваемые движущимися частями механизмов и приборов, могут воздействовать на здоровье человека. Громкие звуки, могут стать причиной проблем со слухом, а длительное воздействие шума более 80 дБ может стать причиной его потери или ухудшения. Чувствительность к монотонным звукам является индивидуальным показателем. Но постоянно повторяющиеся шумы на рабочем месте провоцируют проблемы, связанные с нервной системой и органами слуха.

В данной работе основным источником шума является системный блок ПК, внутри которого работает система охлаждения, состоящая из вентиляторов, воспроизводящих непрерывный шелест или гудение.

Постоянный уровень шума влияет на работоспособность и сосредоточенность человека. Рабочее место соответствует нормам ГОСТ

12.1.003-83 «Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности» [17] и является помещением с минимальным уровнем шума при программировании и разработке программного обеспечения планирования производственных процессов. Кроме того, каждый академический час в работе делается перерыв, который позволяет отключить компьютер и/или выйти из помещения для разгрузки нервной системы и органов слуха.

Характеристикой постоянного шума на рабочем месте является уровень звукового давления, определяемый по формуле:

$$L_a = 20 \lg \frac{P_a}{P_0}. \quad (1.12)$$

Где  $P_a$  – среднеквадратичная величина звукового давления;

$P_0$  – исходное звуковое давление в воздухе, равное  $2 \cdot 10^{-5}$  Па

Замеры звукового давления на рабочем месте показали  $P_a = 385 \cdot 10^{-5}$ .

Согласно формуле 1.12

$$L_a = 20 \lg \frac{385 \cdot 10^{-5}}{2 \cdot 10^{-5}} = 37.73;$$

Предельно допустимые уровни звука на рабочих местах с учётом тяжести труда представлены в таблице 2727.

Таблица 27 – Допустимые уровни звука на рабочем месте

Категория напряженности трудового процесса	Категория тяжести трудового процесса				
	Легкая физическая нагрузка	Средняя физическая нагрузка	Тяжелый труд 1 степени	Тяжелый труд 2 степени	Тяжелый труд 3 степени
Напряженность легкой степени	80	80	75	75	75
Напряженность средней степени	70	70	65	65	65

Напряженный труд 1 степени	60	60	-	-	-
Напряженный труд 2 степени	60	60	-	-	-

Уровень звука, при этом не превышает 40дБ, следовательно, данный уровень шума допустим для лёгкой физической нагрузки и напряжённого труда 2.

### **6.2.3. Повышенный уровень электромагнитных излучений**

Уровень электромагнитных излучений регулируется ГОСТ 12.1.006-84 «Система стандартов безопасности труда (ССБТ). Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля» [19].

В повседневной жизни для людей не заметно воздействия электромагнитных излучений. Мощность источника излучения должна быть достаточно большой, чтобы это отражалось на здоровье и самочувствие организма.

Основные излучающие электромагнитное поле части ПЭВМ – это системный блок, в котором находится процессор, и экран монитора.

На рабочем месте установлены ПЭВМ, оснащённые жидкокристаллическим монитором. Они излучают электромагнитные волны, которые не причиняют человеку вреда, даже при длительной работе.

### **6.2.4. Повышенная напряжённость электрического поля;**

Работа ПЭВМ, кроме электромагнитных излучений сопровождается электростатическим полем, которое может деионизировать окружающую воздушную оболочку. Согласно ГОСТ 12.1.006-84 «Система стандартов безопасности труда (ССБТ). Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля»

[19] для обеспечения безопасности рабочего персонала и оборудования необходимо проводить ионизацию воздуха, путём установки ионизаторов или обеспечения проветривания помещения.

#### **6.2.5. Повышенная или пониженная влажность воздуха;**

Согласно СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания [34] давление, температура и влажность воздуха влияют на здоровье работников, следовательно, они влияют на общее самочувствие, работоспособность и выполнение поставленных задач.

**Рабочее место** из-за своего расположения может быть слишком влажным. Поэтому в стенах аудитории установлена механическая вентиляция, оснащенная вентилятором и отводящая воздух из помещения. Это снижает влажность в помещении. Кроме того, аудитория подключена к системе центрального городского отопления. Это помогает регулировать температуру в период зимних месяцев.

#### **6.2.6. Статические перегрузки;**

В современном мире, почти каждая работа, так или иначе, связана с работой за компьютером. Разработчикам программ, инженерам и всем, кто учится, приходится проводить за ПЭВМ многие часы. При этом пользователь вынужден принимать одну и ту же позу в течение длительного времени, тем самым создавая в работе мышечного корсета статические перегрузки. Неудобная поза, нахождение центра тяжести в одном месте, постоянный наклон вперед вызывают боли в шее и спине.

Работа с ПЭВМ подразумевает обработку большого количества информации. Анализ данных, инженерные исследования, расчеты и разработка программных продуктов требуют высокой концентрации внимания. При работе с визуальной информацией напрягаются глаза, которые

являются зрительными анализаторами человека. Расстояние расположения предмета постоянного визуального контроля не меняется в процессе работы, из-за этого устают глазные мышцы, из-за этого снижается острота зрения. При длительных контактах с дисплеем, постоянного наблюдения схожей по структуре зрительной информации, человек начинает испытывать стресс.

Согласно МР 2.2.9.2311 – 07 «Профилактика стрессового состояния работников при различных видах профессиональной деятельности» [32], разработка программно-алгоритмического комплекса относится к группе В, I категории (до 2х часов) – творческая работа в режиме диалога с ПЭВМ. При выполнении разных групп работ в течение смены за основную принимают такую, которая занимает не менее 50% времени рабочего дня. Для обеспечения оптимальной работоспособности и сохранения здоровья пользователей на протяжении рабочей смены должны устанавливаться регламентированный перерыв, при 8-ми часовом рабочем дне 30 минут. Продолжительность непрерывной работы с ПК не должна превышать 2 часов. Для I категории работ - через 2 часа от начала работы и через 1,5 - 2 часа после обеденного перерыва продолжительностью 15 минут каждый.

Во время регламентированных перерывов с целью сохранения высокой работоспособности выполняется комплекс упражнений. С целью уменьшения отрицательного влияния монотонности целесообразно чередование операций осмысленного текста и числовых данных, чередование редактирования текстов и ввода данных (изменение содержания работы).

Работа над разработкой программного продукта требует сосредоточенности и частого переключения между использованием нескольких программ одновременно. Поэтому для того, чтобы зрительные анализаторы работали на нужном уровне, каждый академический час проводится перерыв в 5-10 минут, а в каждый второй академический час перерыв в 20 минут. Во время перерыва есть возможность выйти из аудитории и выключить на время ПЭВМ.

### **6.2.7. Электробезопасность**

Электробезопасность – система организационных и технических мероприятий и средств, обеспечивающих защиту людей от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества.

Согласно ГОСТ 12.1.030-81 «Система стандартов безопасности труда (ССБТ). Электробезопасность. Защитное заземление. Зануление» [20] помещение, в котором находится рабочее место, относится к категории помещений без повышенной опасности. Его можно охарактеризовать, как сухое, непыльное, с токонепроводящими полами и нормальной температурой воздуха. Температурный режим, влажность воздуха, химическая среда не способствуют разрушению изоляции электрооборудования.

Защита от электрического тока на рабочем месте производится с помощью изоляции токопроводящих частей (все провода изолированы). Электрические устройства, в частности процессор от ПЭВМ расположен в защитном коробе.

Короткое замыкание – это соединение двух точек с разным потенциалом с последующим увеличением тока и выделением большого количества тепла. Вследствие чего короткое замыкание может стать причиной пожара в помещении, при коротком замыкании от электрического тока могут пострадать люди, находящиеся в непосредственной близости от источника возникновения.

На рабочем месте короткое замыкание может быть вызвано либо неисправностью в проводке, либо при работе с компьютером, когда внутри корпуса создается разность фаз и ток может так же повредить всю электросеть.

Для защиты электрической сети от короткого замыкания предусмотрены устройства защитного отключения (УЗО), оснащенные устройствами автоматического отключения – автоматами и

предохранителями. Кроме того, в помещении установлены датчики дыма, которые при возникновении возгорания, вызванного коротким замыканием, оповещают все здание о начавшемся пожаре. Таким образом, рабочее место полностью защищено от возможного короткого замыкания.

### **6.2.8. Статическое электричество**

В рабочем пространстве находится много устройств, которые работают от электрического тока и сделаны из материалов, которые, так или иначе, накапливают на себе статически заряд. Может возникнуть разность потенциалов от статического электричества, и прикосновение человека к заряду может вызвать травмы, ожоги или пожар.

Для защиты оборудования и персонала применяется общее заземление электропроводки и увлажняющие устройства.

## **6.3. Экологическая безопасность**

### **6.3.1. Воздействие на литосферу**

Вышедшее из строя ПЭВМ и сопутствующая оргтехника относится к IV классу опасности и подлежит специальной утилизации. Для оказания наименьшего влияния на окружающую среду, необходимо проводить специальную процедуру утилизации ПЭВМ и оргтехники, при которой более 90% отправится на вторичную переработку и менее 10% будут отправлены на свалки. При этом она должна соответствовать процедуре утилизации ГОСТ Р 53692-2009 Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов [25].

При выходе из строя, а также устаревании компонентов, ПЭВМ и сопутствующая оргтехника начинает представлять собой источник второсортного сырья и относится к IV классу опасности. Каждый ПЭВМ содержит цветные металлы и целый набор опасных для окружающей среды веществ. Это производные газов, тяжелые металлы, среди которых кадмий,

ртуть и свинец. Попадая на свалку, все эти вещества под воздействием внешней среды постепенно проникают в почву.

Люминесцентные лампы при перегорании становятся источником загрязнения. Лампы содержат внутри себя ртуть, которая загрязняет окружающую среду. Кроме того, их корпус состоит преимущественно из стеклянной трубки, которая при неосторожном обращении может разбиться на мелкие осколки.

Документы, перенесенные на бумагу, становятся источником бумажных отходов. Такие отходы медленнее разлагаются из-за предварительной обработки бумаги, а также содержат на себе следы краски, которая попадая в почву ее загрязняет.

Согласно ГОСТ 17.4.3.04-85 «Охрана природы (ССОП). Почвы. Общие требования к контролю и охране от загрязнения» [23] юридические лица имеют право утилизировать оргтехнику только при прохождении процедуры полного списания, подтвержденного актом.

Компания «ТомскСофт» производит сбор отработанных аккумуляторных батарей и передает в утилизацию.

Томский политехнический университет является юридическим лицом, поэтому перегоревшие люминесцентные лампы собираются техническим персоналом, а затем передаются в центр по переработке таких ламп, у которого имеется лицензия на право сбора и переработки люминесцентных ламп. Для макулатуры существуют специально установленные контейнеры, в которые помещаются отработавшие печатные издания, офисная бумага и другие изделия из переработанной целлюлозы. Они отвозятся в пункты по сбору макулатуры, где утилизируются.

## **6.4. Безопасность в чрезвычайных ситуациях**

### **6.4.1. Пожарная безопасность**

Пожарная безопасность может быть обеспечена мерами пожарной профилактики и активной пожарной защиты. Пожарная профилактика включает комплекс мероприятий, направленных на предупреждение пожара или уменьшение его последствий. Возникновение пожара в помещении аудитории может привести к большим материальным потерям и возникновению чрезвычайной ситуации. Чрезвычайные ситуации приводят к полной потере информации и большим трудностям восстановления всей информации в полном объёме.

Согласно ГОСТ 12.1.004-91 «Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Общие требования» [18], рабочее помещение относится к категории В (производства, связанные с обработкой или применением твердых сгораемых веществ и материалов).

В случае возникновения пожара необходимо отключить электропитание, вызвать по телефону пожарную команду, произвести эвакуацию и приступить к ликвидации пожара огнетушителями. При наличии небольшого очага пламени можно воспользоваться подручными средствами с целью прекращения доступа воздуха к объекту возгорания. Покидать помещение необходимо согласно плану эвакуации.

Пожар является чрезвычайной ситуацией для людей, находящихся в помещении. При возникновении пожара срабатывают датчики дыма, которые подадут сигнал общего оповещения всего здания.

На рисунке 2 изображен план эвакуации при возникновении ЧС.



Рисунок 2 – План эвакуации при пожаре

## **Выводы**

В ходе исследования по социальной ответственности были рассмотрены правовые и организационные вопросы обеспечения безопасности, а также организационные мероприятия при компоновке рабочей зоны.

Выделены и проанализированы вредные и опасные факторы, влияющие на человека, находящегося в офисе. Были определены средства защиты, от выявленных вредных и опасных факторов.

Были выявлены бытовые отходы, негативно влияющие на окружающую среду. К ним относятся севшие аккумуляторные устройства, бумажные отходы, перегоревшие лампы. Данные отходы на месте работы утилизируются согласно ГОСТ.

## **Conclusion**

At the beginning of the work, the subject area was studied. Similar software libraries are considered. There are applications with similar technologies in GitHub, but no semantic similar implementations.

Functional requirements are written for the application. In accordance with these requirements, use case, sequence and activity diagrams were constructed. The architecture of the interaction of the system components was also built.

JavaScript programming language was chosen to implement the SDK, as it can be used to create both web and desktop applications. Also, this language has native browser support for the WebSocket protocol, which is important in terms of development requirements.

The core of the library was implemented, which establishes and maintains a connection to the server. Next, the module for building requests to the server and the main external functions of the SDK are implemented.

Every SDK module has been tested and documented. The project is published on GitHub and NPM.

A browser application has been created to demonstrate the capabilities of the SDK.

To conclude, the SDK implementation process includes a full development cycle: obtaining technical specifications, design, development, testing, documentation. As a result of the work, the SDK was designed and created for the web integration of the ManyCam software, which allows other developers to embed ManyCam functions into other applications. Thanks to the developed SDK, relationships between TomskSoft company and clients goes from B2C to B2B level.

## Bibliography

1. API – Wikipedia – URL: <https://ru.wikipedia.org/wiki/API> (дата обращения: 04.03.2021). – Текст: электронный.
2. Builder pattern – Refactoring Guru – URL: <https://refactoring.guru/design-patterns/builder> (дата обращения: 03.03.2022). – Текст: электронный.
3. CSS – Wikipedia – URL: <https://ru.wikipedia.org/wiki/CSS> (дата обращения: 04.03.2021). – Текст: электронный.
4. Fielding, Roy T.; Gettys, James; Mogul, Jeffrey C.; Nielsen, Henrik Frystyk; Masinter, Larry; Leach, Paul J.; Berners-Lee, Tim (June 1999). Hypertext Transfer Protocol – HTTP/1.1. IETF. doi:10.17487/RFC2616. RFC 2616.
5. Flanagan, David (18 April 2011). JavaScript: the definitive guide. Beijing; Farnham: O'Reilly. p. 1. ISBN 978-1-4493-9385-4. – Текст: электронный.
6. GitHub – Wikipedia – URL: <https://en.wikipedia.org/wiki/GitHub> (дата обращения: 15.09.2021). – Текст: электронный.
7. HTML – Wikipedia – URL: <https://ru.wikipedia.org/wiki/HTML> (дата обращения: 10.05.2021). – Текст: электронный.
8. HTTP – MDN – URL: <https://developer.mozilla.org/ru/docs/Web/HTTP> (дата обращения: 02.04.2021). – Текст: электронный.
9. Learn Javascript – WebSocket – URL: <https://learn.javascript.ru/WebSocket> (дата обращения: 01.03.2021). – Текст: электронный.
10. Node.js – URL: <https://nodejs.org/en/> (дата обращения: 12.01.2022). – Текст: электронный.
11. SDK – Wikipedia – URL: <https://ru.wikipedia.org/wiki/SDK> (дата обращения: 04.03.2021). – Текст: электронный.

12. SDK и API: в чем разница? – Хабр – URL: <https://habr.com/ru/company/ibm/blog/572754/> (дата обращения: 10.05.2021). – Текст: электронный.
13. Singleton pattern – Refactoring Guru – URL: <https://refactoring.guru/design-patterns/singleton> (дата обращения: 01.03.2022). – Текст: электронный.
14. Web application – Wikipedia – URL: [https://en.wikipedia.org/wiki/Web\\_application](https://en.wikipedia.org/wiki/Web_application) (дата обращения: 04.11.2021). – Текст: электронный.
15. Библиотека для модульного тестирования – Jest – URL: <https://jestjs.io/ru/docs/getting-started> (дата обращения: 29.03.2021). – Текст: электронный.
16. ГОСТ 12.0.003-2015 Опасные и вредные производственные факторы. Классификация. Перечень опасных и вредных факторов.
17. ГОСТ 12.1.003-83 «Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности».
18. ГОСТ 12.1.004-91 «Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Общие требования».
19. ГОСТ 12.1.006-84 «Система стандартов безопасности труда (ССБТ). Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля».
20. ГОСТ 12.1.030-81 «Система стандартов безопасности труда (ССБТ). Электробезопасность. Защитное заземление. Зануление».
21. ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов.
22. ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования.

23. ГОСТ 17.4.3.04-85 «Охрана природы (ССОП). Почвы. Общие требования к контролю и охране от загрязнения».
24. ГОСТ ISO 9612-2016 Акустика. Измерения шума для оценки его воздействия на человека. Метод измерений на рабочих местах (с Поправкой).
25. ГОСТ Р 53692-2009 Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов.
26. Диаграмма деятельности UML – Планерка – URL: <https://planerka.info/item/diagrammy-deyatelnosti-uml/> (дата обращения: 10.04.2022). – Текст: электронный.
27. Документация – Git – URL: <https://git-scm.com/doc> (дата обращения: 02.03.2021). – Текст: электронный.
28. Документация – JSDoc – URL: <https://jsdoc.app/> (дата обращения: 26.08.2021). – Текст: электронный.
29. Документация – Learn Javascript – URL: <https://learn.javascript.ru/> (дата обращения: 01.03.2021). – Текст: электронный.
30. Документация – NodeJS – URL: <https://nodejs.org/ru/docs/> (дата обращения: 04.03.2021). – Текст: электронный.
31. Модульное тестирование JavaScript и TypeScript в Visual Studio – Docs Microsoft – URL: <https://docs.microsoft.com/ru-ru/visualstudio/javascript/unit-testing-javascript-with-visual-studio?view=vs-2022&tabs=mocha> (дата обращения: 21.07.2021).
32. МР 2.2.9.2311 – 07 «Профилактика стрессового состояния работников при различных видах профессиональной деятельности».
33. Объектно-ориентированное программирование – Wikipedia – URL: <https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0>

%B5\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5 (дата обращения: 16.02.2021). – Текст: электронный.

34. СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания.
35. СП 12.13130.2009 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности;
36. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95.
37. ТОИ Р-45-084-01 Типовая инструкция по охране труда при работе на персональном компьютере.
38. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022).