# **ВВЕДЕНИЕ**

**Актуальность работы.** Экспериментальное обучение – это современный метод исследования психолого-дидактических проблем. На данный момент различается два вида экспериментального обучения: индивидуальный обучающий эксперимент, и коллективное экспериментальное обучение.

Индивидуально обучающий эксперимент существует уже довольно продолжительный срок и основательно вошел в научный процесс. Данный вид эксперимента позволяет определить существующие черты у индивида, а также сформировать новые, направляя обучающегося в процессе исследования. Благодаря экспериментам такого типа можно наблюдать и положительно воздействовать на процесс усваивания и запоминания новой информации.

Коллективное экспериментальное обучение как правило проводится на контрольной группе обучающихся. За контрольную группу берется группа в детском саду, школьный класс, группа студентов, и т.д. Данные исследования проводятся с целью углубленного изучения процесса обучения человека при условии различных, влияющих на процесс обучения, обстоятельств извне для каждого обучающегося.

В совокупности, основной целью экспериментального обучения является выявление наиболее результативного способа усвоения информации с наименьшим количеством негативных последствий и рисков. Внедрение экспериментальных методик обучения или контроля знаний позволяет обеспечить систематичность, регулярность и преемственность необходимого воздействия, а также, дает возможность анализировать полученные данные для дальнейшей обработки и усовершенствования эксперимента.

Так как экспериментальное обучение отличается от обычного процесса обучения, оно вызывает интерес у людей, и получает больший отклик, нежели стандартные обучающие практики. Причем, чем необычнее, но легче и спокойнее эксперимент, тем, как правило, больше обучающихся (или их представителей) соглашаются на подобные методики. Внедрение новых методик образования позволяет разнообразить учебный процесс, давая возможность мозгу человека переключиться от обычного способа получения информации на необычный, что является одним из важных факторов в её успешном запоминании.

Такой способ обучения так же распространяется и в мире информационных технологий. В текущих реалиях IT-технологии прогрессируют с необычайной скоростью, поэтому для них необходимо обучать новых специалистов с использованием актуальных данных. Совместить прогресс в IT-сфере, который шагает широким шагом, и обыденные методики образования с каждым годом всё сложнее. На помощь в таком случае приходит экспериментальное обучение.

В IT-сферу новые методики обучения вводятся постоянно, так как данная область непрерывно растет и развивается. Именно с этим фактором и связана такая популярность экспериментального обучения в мире информационных технологий. Интерактивные модели, новые интересные задачи, обучение поиску нестандартных решений: всё это вводится для того чтобы сделать процесс обучения более продуктивным, захватывающим и приятным.

Не секрет, что обучение с использованием соревновательного процесса дает достаточно хорошие результаты, так как именно желание победить способствует более хорошему получению и усвоению материала у молодого поколения. Для этого и были созданы соревнования Capture the Flag, или, как их сокращенно называют, CTF. CTF – это соревнования по информационным технологиями и информационной безопасности, которые устраиваются ежегодно различными группами людей с целью выявления лучших специалистов в данных областях. Так же, путем участия в данных соревнованиях, обучающиеся стремительно улучшают свои навыки в IT-сфере. Такой прогресс достигается за счет частого чередования процесса поиска новой информации и применения ее на практике.

Для проведения данных соревнований используются специальное ПО, названное CTF-системами или жюрейными платформами. Данное ПО – это набор функций и средств, которые необходимы для корректного проведения соревнований. Так же, каждая система обладает своими достоинствами и недостатками, которые и отличают эти системы друг от друга.

**Предмет исследования:** Жюрейные системы, предназначенные для проведения соревнований Capture the Flag, их функционал, сильные и слабые стороны.

**Цель**: доработка одной из существующих систем путем увеличения количества инструментов системы и улучшением текущего функционала.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Провести обзор методики проведения и жюрейства соревнований.
2. Рассмотреть функционал жюрейных систем.
3. Выявить достоинства и недостатки в нескольких выбранных системах.
4. Выделить систему для доработки.
5. Сформулировать список доработок и внедрить их в систему.
6. Продемонстрировать новый функционал выбранной для доработки системы.

**Методы** используемые в работе: обзор и анализ существующих систем, формирование доработок путем сравнения выбранной системы с существующими, доработка существующего ПО с помощью языка программирования Python и фреймворка Django.

**База** для работы: работа проводилась с помощью анализа существующего ПО для проведения CTF-соревнований, которое находится в свободном доступе в сети Интернет.

**Структура** работы: введение, 4 раздела, заключение, список используемой литературы, приложения.

Во введении работы обоснована актуальность выбранной темы выпускной квалификационной работы, определена цель и задачи работы.

В исследовательском разделе описана методика жюрейства CTF-соревнований, проведен обзор существующих систем, выполнен сравнительный анализ выбранного ПО, выбрана система для доработки, формулирован список доработок для платформы.

В специальном разделе (пока хз че тут будет)

В технологическом разделе (тут будет разработка но хз подробнее че тут будет)

В экономическом разделе (хз че тут будет, че-то экономическое)

В заключении сделаны основополагающие выводы по результатам проведенной работы.

# **ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ**

## Обзор методики проведения и жюрейства CTF-соревнований

Существующие CTF-соревнования бывают двух видов:

* attack defense;
* task-based.

Attack defense это тип соревнований, в котором участникам необходимо получить доступ к серверу противника, при этом удержав оборону своего сервера. Соревнования данного вида не совсем подходят для обучения, как правило, в них участвуют игроки, которые уже имеют достаточно серьезные знания в данной области.

Task-based – это тип соревнований, когда игрокам предоставляется список заданий, которые нужно решить. Решением является флаг, который необходимо ввести в специальное поле. За каждое решенное задание (таск) даются очки, сумма очков за каждый таск разная, зависит она от сложности задачи. Побеждает та команда, которая набирает наибольшее количество баллов за время, отведенное на соревнование. Данный вид соревнований отлично подходит для обучения, так как для решения заданий участникам необходимо самостоятельно добывать пласты недостающих знаний. В связи с этим, работа проведена именно на базе систем для task-based -соревнований.

Процесс проведения состязаний достаточно прост: игроки регистрируются в системе, присоединяясь к одной из команд. Далее, на платформе появляются задания, или, как их еще называют, таски или квесты. На усмотрение организаторов есть два вида отображения квестов:

* команды получают доступ ко всем заданиям сразу, и выбирают задания для решения самостоятельно;
* команды получают доступ к одному или нескольким заданиям, а остальные открываются для команд постепенно, по мере решения ими предыдущих задач.

За каждое решенное задание командам начисляются очки (баллы). Процесс начисления тоже может быть двух видов:

* количество баллов за решение задачи фиксированное, и начисляется командам не зависимо от времени и порядка решения ими задач;
* количество баллов за решение задачи динамически изменяется в зависимости от количества команд, которые решили эту задачу. Т.е. если за определенный промежуток времени задачу решила только одна команда, она получает 100% очков, если их было две, то каждая команда получает по 50% очков, и так далее.

Второй способ начисления очков как правило используется в соревнованиях, где доступ к заданиям открывается постепенно.

Соревнования проводятся определенное количество часов. После того, как время, отведенное на проведение соревнования, выходит, командам закрывается доступ к заданиям. Далее, организаторы состязаний с помощью средств аналитики оценивают, какая команда победила.

В жюрейных системах могут применяться следующие средства аналитики:

* таблицы с результатами очков команд;
* таблицы с количеством решенных командами задач;
* график, отображающий количество решенных командами задач;
* график, отображающий время, затраченное командами на поиск решения задач.
* И т.д.

Выбор, какие средства аналитики использовать в состязаниях, лежит на организаторах соревнований.

## 1.2 Обзор функционала жюрейных систем для соревнований типа tack-based.

Функционал жюрейных систем для соревнований типа tack-based может различаться, есть множество дополнительных функций, которые могут присутствовать в системах, однако, основной перечень необходимых функций, без которых не сможет обойтись ни одна платформа.

1. Редактирование пользователей и команд.

Под редактированием пользователей подразумевается возможность корректировки его анкетных данных, возможность его добавления и удаления, а также зачисления его в определенную команду. Так же, можно редактировать самих команды (название, численность, логотип и т.д.).

Пользователям также присваиваются права (участник/администратор).

Ко всему прочему, организаторам доступна еще и возможность временной блокировки определенного игрока или команды.

1. Редактирование заданий.

Так как основой CTF соревнований являются таски, которые выдаются участникам, у администраторов платформы есть возможность добавления квестов, их удаления, и редактирования их свойств. Также, квесты можно отсортировать по тематике заданий, или по тегам, которые им выставляются.

Не лишней является функция загрузки квеста в систему из архива.

1. Наличие оповещений.

Оповещения от организаторов являются неотъемлемой частью жюрейной системы, с помощью них участники получают уведомления о начале и окончании игры, о появлении новых заданий, информацию о решенных заданиях командой участника. Кроме оповещений для всех игроков и для конкретной команды, есть оповещения для конкретного участника или нескольких участников. Такая функция очень удобна в случае, когда организатору необходимо донести информацию до одного или нескольких участников, потому что не нужно ставить в известность всю команду или всех игроков.

1. Установка времени начала и окончания игры.

Данный пункт один из самых важных: каждое соревнование проходит в четко оговоренных временных рамках, и чтобы не допустить возможность досрочного получения доступа к заданиям (или, наоборот, поздней сдачи), для каждого этапа соревнований выставляется время начала и окончания. До и после этого отрезка времени все задания этапа можно заморозить.

1. Наличие панели администратора.

Для организаторов соревнований администрирование большого количества команд не должно составлять большой труд. По возможности, панель администратора делается максимально удобной, функциональной и интуитивно понятной.

1. Просмотр аналитики соревнований.

Для наблюдения за участниками в режиме on-line в системах присутствуют средства аналитики. Они позволяют осуществлять контроль за скоростью решения заданий, просмотр успехов отдельно взятой команды или участника. Аналитика обычно выстраивается в виде графика.

## 1.3 Обзор существующих жюрейных систем

# 1.3.1 Обзор жюрейной платформы «CTFd»

CTFd – самая распространенная платформа для соревнований Capture The Flag. Она ориентирована на простоту использования и расширенную кастомизацию. В состав платформы входит всё необходимое для развертывания, настройка системы происходит с помощью множества тем.

CTFd – продукт группы разработчиков из США. Это полностью открытое программное обеспечение, его исходный код расположен на GitHub. Данную систему можно самостоятельно развернуть на хостинге, однако, для потребителей, которые не хотят заниматься такими вещами как настройка и развертывание, команда CTFd может предложить вариант «под ключ»: арендовать хостинг с развернутой на нем платформой за определенную сумму в месяц (в зависимости от перечня функций хостинга).

Платформа имеет множество функций, начиная от основных, описанных в предыдущем разделе, и заканчивая уникальными только для неё.

Из дополнительных функций можно перечислить:

1. Автоматическую защиту от brute force.

Данная особенность системы очень полезна для соревнований типа tack based, так как участники могут начать «перебирать» ответы в поисках верного.

1. Автоматическое скрытие заданий.

Эта функция срабатывает тогда, когда у участника (или у команды) закончились попытки для решения задания.

1. Временное замораживание статистики.

Может использоваться организаторами соревнований для различных целей: например, для того чтобы команды не знали предварительные результаты своих соперников.

1. Отложенная установка временных рамок соревнования.

Обеспечивает удобство для организаторов. Не обязательно нажимать на кнопку для старта состязаний - таймер сработает в определенный час и соревнование начнется автоматически.

1. Использование языка разметки Markdown.

Данная особенность упрощает управление системой, давая возможность легко редактировать контент на страницах.

1. Интеграция с MajorLeagueCyber.

Данная система тесно связана с MajorLeagueCyber (MLC). MLC – это средство отслеживание статистики для CTF, позволяющее отслеживать и планировать события.

1. Возможность переключения способа скорринга (начисления очков) с статического на динамический.

Как было уже ранее сказано в п.1.1 исследовательского раздела, существует два способа начисления очков. И CTFd дает возможность организаторам переключаться между этими способами.

1. И другие.

Установить систему можно несколькими путями. Разработчики в первую очередь советуют использовать Docker, так как это самый простой и удобный способ. Но если такой вариант не подходит, можно использовать сервер отладки Flask.

Для удобства использования разработчиками составлена краткая и подробная инструкция по развертыванию и использованию CTFd.

Как бы хороша система не была, в ней всегда будут какие-то недостатки. К недостаткам системы CTFd можно причислить:

1. (нужно спросить, какие недостатки у системы)

1.3.2 Обзор жюрейной системы «Project Asya»

Project Asya – система для проведения CTF-соревнований, произведенная на свет российскими разработчиками. Его исходный код расположен в репозитории на GitHub, как и у предыдущей разработки.

В отличие от CTFd, Project Asya не является популярной жюрейной системой, которой пользуются тысячи пользователей, она малопопулярна, но не уступает по производительности своим аналогам. Продукт был создан разработчиками команды Keva (имя система унаследовала от одной из участниц команды), и использовался для проведения соревнований UralCTF.

У этой системы есть ряд преимуществ, благодаря некоторым из них она даже может обойти свои более известные аналоги:

1. Простой и удобный интерфейс.

Это было отмечено как и организаторами соревнований UralCTF, так и его участниками. Систему все чаще выбирают для проведения состязаний из-за её интуитивно понятного интерфейса и незагруженности.

1. Простая система мониторинга

Является несомненным преимуществом в проведении соревнований. Бонусом к данному пункту является то, что система мониторинга динамичная, и позволяет показывать прогресс участников в режиме online.

1. Быстрая установка.

Project Asya показывает отличные результаты в развертывании: процесс запускается всего одной командой и длится всего две минуты (что подтвердили организаторы соревнований).

1. Стабильность.

Является главным критерием любой системы: так как система поддается большой нагрузке, она должна уметь справляться с неравномерным напряжением, при этом не теряя в производительности.

1. Доступность.

Как уже было сказано выше, Project Asya совершенно бесплатна и доступна всем желающим для использования. Эта особенность дает возможность тренироваться в развертывании и управлении системой задолго до проведения собственных соревнований.

1. И другие.

Эту систему, как и любую другую, не обошли стороной недостатки. Их у данного проекта, к счастью, немного:

1. Отсутствие демонстрационной версии

Так как Project Asya была сделана не для коммерческого использования, и в основном её разворачивают как раз те же люди, которые ее разрабатывали, в демонстрационной версии для них она не нуждается. Но стороннему обывателю перед развертыванием системы было бы полезно увидеть ее демо-вариант, чтобы понять, подходит она или нет.

1. Малопопулярность

Платформа не так известна, как ее аналоги, и это может создать неудобства. К ней есть краткая инструкция, однако, она очень малоинформативна. Да, разработчики системы заявляют, что, при необходимости, они могут оказать помощь и поддержку в работе с системой, однако, отсутствие отзывов о продукте, а также наличие более подробной инструкции немного омрачает вышеперечисленные плюсы.

Тем не менее, будучи малоизвестной системой, Project Asya имеет очень хорошую поддержку в лице своих разработчиков. Во время разработки и доведения до совершенства данного проекта, система получала регулярные обновления, которые можно отследить по обновлениям на GitHub. Платформа была спроектирована по аналогии со своими зарубежными конкурентами, с ориентиром на простоту и удобство, но она не лишена своей уникальности и изящности.

1.3.3 Обзор жюрейной системы от Hackerdom

Hackerdom – русская команда, образовавшаяся в 2005 году на математико-механическом факультете Уральского государственного университета. Она развивается и по сей день, и в неё входят студенты самых разных направлений и курсов.

Команда занимается изучением компьютерной безопасности, организацией и участием в соревнованиях CTF. Поначалу, молодые хакеры только участвовали в соревнованиях, но затем начали понемногу организовывать свои. В настоящее время студенты организовывают всероссийские и международные соревнования QCTF, RuCTF и RuCTFE. Так же, Hackerdom проводит курсы для тех, кто хочет играть в CTF, но не обладает достаточными знаниями. Участие в курсах абсолютно бесплатно и не требует сильно специфических и углубленных знаний.

Использование сторонней жюрейной системы для соревнований перестало подходить команде, поэтому вскоре студенты создали свою собственную.

Основными разработчиками системы стали Михаил Гнедашев и Александр Гейн.

Интерфейс системы прост и незамысловат: его можно отредактировать под нужды участников, но изначально он сделан в черно-белых тонах. Нерешенные задания выделены синим цветом, решенные – зеленым. Есть доступ к своему профилю, квестам, а также к турнирной таблице.

Платформа от Hackerdom находится в открытом доступе, её исходный код выложен в репозитории на GitHub. Там же находятся и модификации системы для различных проведенных соревнований.

Одним из самых главных плюсов данной системы является то, что её можно протестировать, не раскатывая на своем сервере. Это дает ей неоспоримое преимущество перед остальными системами, так как все мы с большей вероятностью отдадим предпочтение тому сервису, который можем самостоятельно опробовать без лишних усилий. На сервис уже добавлено очень большое количество заданий (с прошлых CTF), которые используются в обучении школьников старших классов и студентов.

К другим полезным функциям системы можно отнести:

1. Стабильность

Была отмечена как разработчиками, так и участниками. За всё время проведения множества соревнований, система постоянно улучшалась, и с каждым годом становилась всё более стабильной.

1. Интуитивность в использовании

Все функции системы просты и понятны, она не нагружена лишними малоиспользуемыми возможностями ПО. Одновременно с этим, в ней присутствуют главные опции, перечисленные выше, что делает её приятной для использования.

1. Поддержка разработчиков

Несомненно, наиболее успешно используется та система, разработчики которой поддерживают свой продукт. Детище Hackerdom регулярно обновляется, для каждых соревнований выпускается своя, немного модифицированная, полноценная версия системы, которую потом разработчики любезно выкладывают на GitHub. Там же находятся и задания с прошедших соревнований, который можно решать, тем самым тренируясь, и даже использовать в своих собственных состязаниях.

1. Доступность.

Аналогично предыдущей системе, данная платформа обладает свободной лицензией и может свободно разворачиваться любыми пользователями.

1. И другие.

Как всегда, не обошлось и без недостатков. Их у системы немного:

1. Отсутствие инструкции по развертыванию

Несмотря на то, что система открытая и довольно известная, у неё отсутствует даже краткая информация о том, как правильно её установить. Это создает небольшие неудобства, так как новичку в данной области порой приходится довольно долго разбираться с настройкой.

1. (еще какой-нибудь недостаток)

Невзирая на недостатки, жюрейная система от Hackerdom является популярной и хорошей системой, которые пользуются многие организаторы для проведения соревнований.

1.3.4 Обзор жюрейной системы от Facebook

Facebook это популярная социальная сеть, основанная в 2004 году. Позже, так стала называться и владеющая ею компания. Поначалу, дальше сайта для общения между людьми компания не двигалась, но потом она разрослась, начав занимать всё большие IT-сферы. Так она поглотила несколько крупных сервисов, таких как Instagram и What’sApp.

Параллельно с развитием основной компании и поглощением дочерних, Facebook занялся развитием в сфере проведения соревнований Capture the Flag. Изначально соревнования проводились только на уровне колледжа, а целью компании было помочь детям школьного возраста открыть для себя мир информационных технологий и компьютерной безопасности. Со временем аудитория на соревнованиях расширилась, и Facebook стал организовывать соревнования уже по всему миру.

Все состязания уже с самого начала проводились компанией на специальной CTF-платформе, которая был разработана в Facebook для этой цели. Она сочетала в себе преимущества других платформ, ловко избегая популярных недостатков, а дополнительно обладала уникальным дизайном, который не был замечен больше ни на одной жюрейной системе.

Как и у всех других платформ, у системы от Facebook (сокращенно ее называют FBCTF) есть стандартные функции, такие как:

* Регистрация команд и участников
* Размещение информации о заданиях
* Система подсказок для участников
* И т.д.

Из преимуществ платформы от Facebook можно выделить:

1. Уникальный интерфейс

Такого интерфейса платформы, как у FBCTF больше нет нигде. Будучи платформой от разработчиков популярной социальной сети, Facebook смог удивить пользователей красотой и практичностью системы. В отличие от других жюрейных систем, у FBCTF тёмный интерфейс в неоновых тонах (преобладает чёрный, индиго и красный). Сами задания размещаются не в простой табличке, а на карте мира: каждый квест находится в своей стране. На этой же карте можно увидеть, какие еще команды решили задания. По бокам от карты, как в популярных играх, находятся меню с список других команд, списком решенных заданий и многим другим.

1. Доступность

FBCTF является бесплатным продуктом. Весь его исходный код выложен на GitHub.

1. Наличие демонстрационной записи

Facebook хорошо позаботились о том, чтобы у пользователей их платформы было четкое представление о продукте. Поэтому, для краткого ознакомления с интерфейсом и возможностями FBCTF разработчики сделали, пусть и короткую, но довольно информативную видеозапись использования системы.

1. Подробная инструкция по установке

Немногие платформы для проведения CTF-соревнований могут похвастаться наличием инструкции по установке для пользователей. Да и обычно она довольно короткая. Facebook в этом плане превзошел своих конкурентов и для своей платформы сделал подробную инструкцию по установке, которая прилагается к информационному файлу в репозитории на GitHub.

Несмотря на все преимущества системы, её так же не обошли стороной недостатки:

1. Необходимость высокопроизводительных комплектующих

В отличие от других систем, FBCTF является довольно требовательным ПО, что является обратной стороной медали для всех вышеперечисленных достоинств.

1. Усложненная установка.

В отличие от своих конкурентов, у FBCTF установка производится не в «два клика» и не за «две минуты». Однако, она всё равно не является слишком сложной для пользователей.

В целом, FBCTF довольно сильно отличается от своих конкурентов, начиная интерфейсом и заканчивая кодом. Тем не менее, он является довольно хорошей платформой для проведения соревнований Capture the Flag.

1.3.5 Обзор жюрейной системы SchoolCTF

Платформа для проведения CTF-соревнований SchoolCTF была создана русскими разработчиками в 2014 году для проведения конкурса SchoolVolgaCTF среди учеников школ. Данный продукт бесплатный и свободно распространяемый, его исходный код выложен на ресурсе GitHub

Данное ПО задумывалась как легковесная и простое в управлении жюрейная система, которую под силу развернуть любому человеку, даже тому, у кого нет опыта в проведении подобных состязаний. SchoolCTF является не самой известной платформой, тем не менее она обладает рядом преимуществ:

1. Простота интерфейса

Система действительно несложная и удобная в использовании как организаторами, так и участниками. Интуитивно понятный интерфейс, малое количество разделов, отсутствие излишней нагруженности делают ее подходящим вариантом для проведения соревнований среднего масштаба.

1. Доступность

Данная платформа является полностью открытым ПО, его исходный код выложен на GitHub

(хз какие тут еще есть достоинства)

Данную систему, как и любую другую, не обошли стороной недостатки:

1. отсутствие иных видов аналитики кроме табличных.

В данной жюрейной системе из способов аналитики соревнований существуют только таблицы с результатами решения участниками задач. Это не совсем удобно для организаторов соревнований.

1. Отсутствие выбора скорринга

В данной системе, как и в нескольких других существующих системах, отсутствует выбор способа начисления очков. Это делает соревнования более однообразными от раза к разу.

1. Отсутствие выбора способа открытия заданий.

В данной системе отсутствует функция отображения квестов последовательно, что так же делает соревнования скучнее.

1. Отсутствие отображения дополнительной информации в заданиях.

Под дополнительной информацией подразумеваются второстепенные данные о задачах (количество команд, которые решили квест, автор задания и т.д.). Этот недостаток лишает участников соревнований полного представления о задании, а значит, не дает возможности выстроить более достоверную стратегию игры.

1. Отсутствие возможности отображения всплывающих уведомлений с произвольным текстом.

Из всплывающих уведомлений в списке своих опций SchoolCTF имеет только уведомления от самой платформы (информация о неверном логине и пароле, уведомления об ошибке в системе и т.д.). Соответственно, организаторы соревнований не имеют возможности внутри системы оповестить участников о каких-либо изменениях или новой информации в игре.

# 1.3.6 Итоговая сравнительная таблица функционала обозреваемых систем

В таблице 1 приведен список жюрейных систем (первый столбец таблицы) и список функций (первый столбец таблицы) для последующего выбора системы для доработки:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CTFd | Project Asya | Жюрейная платформа от команды Hackerdom | FBCTF | SchoolCTF |
| Система свободно распространяемая | да | да | да | да | Да |
| Простота интерфейса | да | да | да | нет | да |
| Наличие инструкции по установке | да | нет | нет | да | нет |
| Различные виды аналитики (таблицы, графики и т.д.) | да | нет | да | да | нет |
| Динамический скорринг | да | нет | нет | да | нет |
| Отображение доп. информации в заданиях | да | да | да | да | нет |
| Наличие всплывающих уведомлений | да | да | да | да | нет |

Таблица 1. Сравнение функционала обозреваемых жюрейных систем (узнать каким шрифтом форматируются подписи к таблицам)

Путем сравнения выбранных жюрейных систем, можно заметить, что больше всего по количеству недостатков лидирует жюрейная платформа SchoolCTF. Таким образом, именно её решено выбрать в качестве системы для доработки в рамках данной работы.

# **1.4 Список планируемых доработок для жюрейной системы SchoolCTF**

Путем сравнения системы SchoolCTF с её аналогами, было выявлено, что в платформе отсутствуют функции, которые, в большинстве своем, присущи её аналогам. Исходя из данного факта, для доработки были кратко сформулированы следующие опции:

1. добавление возможности динамического скорринга;
2. отображение дополнительной информации задачах:

2.1 имя автора задачи;

2.2 рейтинг сложности задачи;

2.3 список названий команд, решивших задачу;

1. ввод аналитики в виде графиков;
2. составление краткой инструкции по разворачиванию системы;
3. отображение всплывающих уведомлений с текстом, заданным организаторами;

Так же, для повышения функциональности и уникальности платфомы, были выбраны следующие опции для доработки:

1. включение и отключение возможности регистрации через панель администратора;
2. выбор последовательности отображения заданий для участников соревнований.

Таким образом, путем сравнения различных жюрейных систем, был сформулирован список опций для доработки в функционале в выбранной для этой цели платформе.

# **1.5 Описание планируемого результата**

В результате выполнения цели данной работы планируется осуществление доработок системы SchoolCTF, описанных в п. 1.3 выпускной квалификационной работы.

Наглядно проиллюстрировать планируемый функционал жюрейной платформы можно с помощью use-case диаграммы (рисунок 1):



Рисунок 1. USE-CASE диаграмма жюрейной системы SchoolCTF

# СПЕЦИАЛЬНЫЙ РАЗДЕЛ

## 2.1 Описание выбранного ПО для разработки (название переделать)

2.1.1 Выбор языка программирования для разработки.

К выполнению поставленной задачи можно подойти двумя путями:

* Переписать с нуля имеющийся проект
* Использовать готовую основу имеющегося проекта для последующих доработок

Рассмотрим первый вариант выполнения поставленной задачи.

Для качественного выполнения поставленной задачи необходим гибкий, современный язык программирования, который осуществляет поддержку следующих пунктов:

* серверных фреймворков для разработки сайта;
* использования HTTP-серверов;
* написания HTML-страниц с нуля.

Рассмотрим несколько подходящих языков программирования, являющихся самыми популярными для данной задачи в настоящее время:

* Java

Java — язык программирования, используемый, в основном, для разработки родных Android-приложений для смартфонов и планшетов, а так же для разработки ПО для ОС Windows. Популярность Java у разработчиков связана с простотой и надежностью языка, который обеспечивает долгосрочную совместимость написанных на нём продуктов. Для разработки web-страниц Java поддерживает интерфейс под названием Servlet. Он взаимодействует с клиентом посредством принципа запрос-ответ. Для реализации HTML-страниц в Java используется технология jsp, задача которой динамически генерировать веб – страницы. Она имеет как статические, так и динамические компоненты. Статической частью jsp являются исходные данные, которые сохранены в одном из текстовых форматов (обычно HTML). Краткая схема работы веб-программы на Java выглядит следующим образом:



Рисунок 2. Краткая схема работы веб-приложения, написанного на Java

* PHP

Язык программирования PHP был сконструирован специально для веб-разработки, что является его главным достоинством. Его код можно встраивать напрямую в HTML-страницу. Он отделяется от основного кода HTML-страницами специальными начальными и конечными тегами. Каждый PHP-скрипт обрабатывается на сервере и генерирует HTML-страницу, которая отправляется клиенту. Данный язык программирования является очень простым в освоении, однако, большинство действий, которые в других языках можно не писать «руками» в PHP необходимо прописывать вручную.

Для того чтобы PHP-скрипты корректно работали, необходимо подключать сервер к проекту. Как правило, разработчики подключают готовую сборку из сервера, самого языка PHP, и базы данных. Хорошим примером такого случая может служить Denwer – набор, где содержатся все инструменты для веб-разработчика.

Краткая схема работы сайта, написанного на PHP выглядит так (рисунок 3):



Рисунок 3. Краткая схема работы сайта, написанного на PHP

В качестве серверного фреймворка PHP поддерживает такой продукт как Zend Framework, который свободно распространяется и основывается на принципах MVC (Model-View-Controller). MVC – это схема разделения данных (архитектурный шаблон) пользовательского приложения таким образом, что каждый из трех его компонентов (модель, представление, контроллер) может модифицироваться независимо друг от друга.

Процесс работы архитектурного шаблона можно проиллюстрировать следующим изображением:



Рисунок 4. Иллюстрация работы архитектурного шаблона MVC

MVC является популярной и современной схемой разделения данных, которая используется в большом количестве фреймворков.

* Ruby

Ruby – это язык программирования, в котором соединяются части таких языков как Perl, Smalltalk, Lisp, Eiffel и Ada для того чтобы обеспечивать баланс между парадигмой функционального программирования и принципами императивного программирования. Данный язык, по словам его автора, «прост снаружи, но сложен внутри, совсем как человек». В данном языке всё является объектами. Его создатель оперировал убеждением о необходимости более мощного языка, чем Perl и более объектно-ориентированного, чем Python. Ruby так же очень гибкий и позволяет свободно менять его части, он отличается высокой переносимостью и работает на большинстве современных ОС.

Для разработки веб-приложений Ruby поддерживает фреймворк под названием Ruby on Rails. Он обеспечивает интеграцию с веб-сервером, сервером баз данных, является открытым ПО. Как и Zend Framework, Ruby on Rails реализует архитектурный шаблон MVC.

* Python

Python – это высокоуровневый язык программирования, который прежде всего ориентирован на простоту, читаемость кода, и повышение производительности разработчиков. Данный язык поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Он активно развивается, новые его версии выходят примерно раз в два года. Python не подвергался официальной стандартизации. Его код понятен, удобен и прост в изучении.

Для реализации веб-приложений на языке Python существует фреймворк Django. Это свободно распространяемое ПО. Фреймворк, как и его аналоги на других языках программирования, использует шаблон проектирования MVC. Однако, в Django есть существенные отличия от своих аналогов: сайты на нём строятся из одного или нескольких приложений, которые рекомендуется делать подключаемыми. Так же, обработчики URL в Django конфигурируются с помощью регулярных выражений, что более удобно.

В Django присутствует собственный ORM (объектно-реляционное отображение), что позволяет фреймворку генерировать собственную схему базы данных, не прибегая к сторонним разработкам, а так же защита от SQL-иньекций. Ко всему прочему, у фреймворка имеется уже готовый веб-интерфейс панели администратора, что очень облегчает и ускоряет работу.

**Вывод:** для реализации поставленной задачи необходим простой язык программирования, который может работать с фреймворками. Java для решения данной задачи слишком сложен в исполнении: для того чтобы отобразить даже одну веб-страницу необходимо совершить достаточно много действий.

SchoolCTF написан на Python с использованием Django. Ruby и PHP так же подходят для доработки этой жюрейной системы, однако, для того, чтобы использовать любой из этих языков программирования, придется переписывать проект с нуля. Поэтому было решено выбрать язык программирования Python: благодаря использованию именно его в проекте, нет необходимости переписывать проект заново под другое ПО, к тому же, проект реализован на Django, который, как было сказано выше, выделяется среди своих аналогов тем, что он более гибок, удобен, безопасен, и у него есть свой веб-интерфейс панели администратора.

Таким образом, был выбран второй вариант решения поставленной задачи: использовать готовую основу имеющегося проекта для последующих доработок.

* + 1. Выбор СУБД для разработки

Система управления базами данных (СУБД) играет одну из ключевых ролей в проекте. В рамках выполнения поставленной задачи главным ее критерием является удобство в обращении с ней и стабильность в работе.

Ниже перечислены три популярных базы данных, которые могут подойти для выполнения данной задачи:

* MySQL

MySQL является свободно распространяемой реляционной СУБД от компании Oracle. В основном используется для малых и средних приложений. Входит в состав некоторых серверов, а так же в портативную сборку Denwer. Достоинствами данной СУБД являются её гибкость, поддержка большого количества видов таблиц, а так же наличие примеров таблиц (EXAMPLE ) в составе СУБД, которые демонстрируют принципы создания новых таблиц.

* PostgreSQL

PostgreSQL – это свободно распространяемая объектно-реляционная СУБД. Используется для средних и больших приложений. Имеет ряд преимуществ перед своими конкурентами (например, перед MySQL), так как то, что СУБД объектно-реляционная дает возможность сохранять такие данные как многомерные массивы или сетевые адреса без использования обходного пути в виде дополнительно й таблицы. Так же, PostgreSQL поддерживает JSON в полной мере, в отличие от других СУБД, что обеспечивает дополнительную гибкость в работе.

* SQLite

SQLite – это компактная встраиваемая СУБД. Под понятием «встраиваемая» подразумевается то, что движок данной СУБД является не отдельно работающим процессом, а библиотекой, с которой взаимодействует программа. Поэтому для обмена информацией используются вызовы функций (API). Этот способ гораздо проще, удобнее и быстрее. Так же, преимуществом данной СУБД является динамическое типизирование данных.

Хоть SQLite и обладает более скудным функционалом, чем её полноценные аналоги, она вполне подходит для работы с малыми и средними приложениями. Она так же используется многими популярными программными продуктами для хранения данных (такими как ОС Android или браузер Google Chrome), привязана ко многим языкам программирования и входит в состав фреймворков Qt и Django.

**Вывод**: для реализации поставленной задачи по минимальным требованиям подходят все три приведенные СУБД. Однако, СУБД MySQL и PostgreSQL более сложные в использовании, и могут быть более нестабильны в работе, так как должны подключаться «со стороны». Так как в состав фреймворка Django уже входит SQLite, предпочтение в используемой СУБД было отдано этому варианту.

2.1.3 Выбор среды разработки

Для выполнения поставленной задачи необходима среда разработки, которая будет удовлетворять следующим требованиям:

* поддержка языка Python;
* удобная работа с фреймворком Django;
* прямое взаимодействие с распределенной системой контроля версий (Git);
* быстрый запуск проекта;
* запуск кода из среды разработки;
* поддержка отладки.

Всем этим требованиям удовлетворяет полнофункциональная интегрированная среда разработки (IDE) под названием PyCharm. Данная IDE так же выделяется хорошей поддержкой, регулярными обновлениями и возможностью работы прямо «из коробки».

В связи с вышеперечисленными достоинствами, PyCharm был выбран для реализации поставленного задания.

## 2.2 Объектное моделирование жюрейной системы SchoolCTF

2.2.1 Обоснование технологии проектирования

Для успешной реализации проекта объект проектирования (ИС) должен быть прежде всего адекватно описан, должны быть построены полные и непротиворечивые функциональные и информационные модели ИС. Для этих целей применяются CASE-технологии, которые представляет собой методологию проектирования ИС, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех этапах разработки и сопровождения ИС и разрабатывать приложения в соответствии с информационными потребностями пользователей.

Для разработки информационных систем также проводят моделирование бизнес процессов, целью которого является систематизация знаний о компании и ее бизнес-процессах в наглядной графической форме более удобной для аналитической обработки полученной информации. Моделирование бизнес-процессов позволяет проанализировать не только, как работает предприятие в целом, как оно взаимодействует с внешними система, но и как организована работа в каждом отдельно взятом модуле системы.

На этапе структурного моделирования в модели должны быть отражены:

* существующая организационная структура;
* документы и иные сущности, используемые при исполнении моделируемых бизнес-процессов и необходимые для моделирования документооборота, с описаниями их основного смысла;
* структуру бизнес-процессов, отражающую их иерархию от более общих групп к частным бизнес-процессам;
* диаграммы взаимодействия для конечных бизнес-процессов, отражающие последовательность создания и перемещения документов (данных, материалов, ресурсов и т.п.) между действующими лицами.

Для отображения основных частей информационной системы используют UML - язык для визуализации, специфицирования, конструирования и документирования артефактов программных систем. Язык состоит из словаря и правил, позволяющих комбинировать входящие в него слова и получать осмысленные конструкции. В языке моделирования словарь и правила ориентированы на концептуальное и физическое представление системы. Язык моделирования, подобный UML, является стандартным средством для составления описания бизнес процессов.

Моделирование необходимо для понимания системы. При этом единственной модели никогда не бывает достаточно. Напротив, для понимания любой нетривиальной системы приходится разрабатывать большое количество взаимосвязанных моделей. В применении к программным системам это означает, что необходим язык, с помощью которого можно с различных точек зрения описать представления архитектуры системы на протяжении цикла ее разработки.

В связи с основными требованиями, предъявляемые к модели информационной системы разработана следующая технология проектирования:

* разработать диаграмму вариантов использования, на которой отобразить отношения, существующие между актерами и прецедентами;
* разработать диаграмму последовательности, на которой показать взаимодействия объектов, упорядоченных во времени;
* разработать диаграмму классов, с помощью которой описать структуру системы, показав ее классы, их атрибуты и операторы, а также взаимосвязи этих классов.

Под выбранную технологию проектирования наиболее эффективным является выбор CASE-средства Draw.io

Draw.io – это веб-сервис, предназначенный для формирования диаграмм и схем. Draw.io работает в браузерах, поддерживающих HTML5. В нем доступны базовые и продвинутые возможности построения схем и диаграмм, а так же синхронизация с Google-диском и возможность конвертации построенной схемы в формат изображений (JPEG, PNG и др.).

2.2.2 Диаграмма последовательности (sequence diagram)

Диаграмма последовательности отражает поток событий, происходящих в рамках варианта использования.

В диаграммах последовательности действий взаимодействие объектов в системе происходит посредством приема и передачи сообщений объектами-клиентами и обработки этих сообщений объектами-серверами. При этом в разных ситуациях одни и те же объекты могут выступать и в качестве клиентов, и в качестве серверов.

На диаграмме последовательности изображено упорядоченное во времени взаимодействие объектов при отправке участников ответа на задание.

Основным содержимым диаграммы являются сообщения, которые изображаются горизонтальными стрелками, переходящие от одной вертикальной линии жизни объекта к другой. Соответствующее сообщение ассоциируется с некоторым действием внутренних объектов: базой данных, сервером, пользователем, графическим интерфейсом. Каждое сообщение обладает именем, рядом передаваемых параметров, а также возвращаемым значением.

Диаграмма последовательности жюрейной системы SchoolCTF представлена на рисунке 5.



Рисунок 5. Диаграмма последовательности (sequence diagram)

Так как диаграмма последовательности описывает динамические аспекты жюрейной системы SchoolCTF, то можно выделить следующие действия, протекающие в течение линии жизни объектов:

* отправка ответа на сервер;
* проверка информации о правильности решения;
* выдача ответа;
* внос изменений в базу данных.

**Отправка ответа на сервер** происходит путем ввода участником правильного ответа в специализированное окно в веб-интерфейсе платформы.

**Проверка информации о правильности решения** происходит путем запроса сервера к базе данных. Сервер запрашивает у базы данных верный вариант ответа на задачу, база данных его выдает, сервер сверяет введенный участником ответ и ответ, который получен от базы данных.

**Выдача ответа** так же проводится сервером, плавно перетекая из проверки информации о правильности решения. Если решения участника было верным, сервер передает данную информацию в веб-интерфейс, который отображает её участнику. Если же ответ был неверный, то сервер так же передает информацию о неверности ответа в веб-интерфейс, который далее отображает её участнику.

**Внос изменений в базу данных** происходит в случае ввода участником правильного ответа на задание. В данной ситуации сервер перед отправкой сообщения участнику делает запрос к базе данных на внос изменений в список решенных участником заданий. После того, как он получает утвердительный ответ о том, что изменения внесены успешно, он оповещает участника о правильности его решения.

2.2.3 Диаграмма деятельности (activity diagram)

Диаграмма деятельности — UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

Диаграммы деятельности состоят из ограниченного количества фигур, соединённых стрелками. Основные фигуры:

* Прямоугольники с закруглениями — действия
* Ромбы — решения
* Широкие полосы — начало (разветвление) и окончание (схождение) ветвления действий
* Чёрный круг — начало процесса (начальный узел)
* Чёрный круг с обводкой — окончание процесса (финальный узел)
* Стрелки идут от начала к концу процесса и показывают потоки управления или потоки объектов (данных).

На данной диаграмме (рисунок 6) представлены действия, выполняемые пользователем и системой при отправке участником ответа на задание в системе.



Рисунок 6. Диаграмма деятельности

Диаграмма содержит следующие наборы действий, образующие деятельность:

* отправка ответа на задачу в систему;
* отправка запроса на выдачу результата на сервер;
* отправка запроса в базу данных;
* выдача правильного ответа базой данных;
* сверка введенного и правильного ответов:
* ответ верный;
* ответ неверный;
* отправка запроса на изменение данных в базу данных;
* вывод сообщения участнику.

2.2.4 Диаграмма классов (class diagram)

В программной инженерии диаграмма классов в унифицированном языке моделирования (UML) представляет собой тип статической структурной схемы, которая описывает структуру системы, показывая классы системы, их атрибуты, операции (или методы), а также отношения между объектами.

Диаграмма классов является основным строительным блоком объектно-ориентированного моделирования. Он используется как для общего концептуального моделирования системного приложения, так и для детального моделирования перевода моделей в программный код. Диаграммы классов также могут использоваться для моделирования данных. Классы в диаграмме классов представляют, как основные элементы, взаимодействия в приложении, так и классы, которые будут запрограммированы.

При проектировании системы несколько классов идентифицируются и группируются вместе в диаграмме классов, которая помогает определить статические отношения между ними. При детальном моделировании классы концептуального проектирования часто разбиваются на несколько подклассов.

Для дальнейшего описания поведения систем эти диаграммы классов могут быть дополнены диаграммой состояний или машиной состояний UML.

(А диаграммы тут пока и нет, она будет, но чуть позже, когда я посмотрю как изнутри выглядит приложение)

2.2.5 Диаграмма компонентов (component diagram)

Диаграмма компонентов — элемент языка моделирования UML, статическая структурная диаграмма, которая показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п.

С помощью диаграммы компонентов представляются инкапсулированные классы вместе с их интерфейсными оболочками, портами и внутренними структурами (которые тоже могут состоять из компонентов и коннекторов).

Компоненты связываются через зависимости, когда соединяется требуемый интерфейс одного компонента с имеющимся интерфейсом другого компонента. Таким образом иллюстрируются отношения клиент-источник между двумя компонентами.

Зависимость показывает, что один компонент предоставляет сервис, необходимый другому компоненту. Зависимость изображается стрелкой от интерфейса или порта клиента к импортируемому интерфейсу.

Когда диаграмма компонентов используется, чтобы показать внутреннюю структуру компонентов, предоставляемый и требуемый интерфейсы составного компонента, могут делегироваться в соответствующие интерфейсы внутренних компонентов.

(А она тоже тут появится после того, как я посмотрю, как изнутри выглядит приложение, чтобы не переделывать её потом).