

# Техническое задание на разработку информационной системы

Калинина А. А.

## Оглавление

<b>1. Общие сведения</b>	<b>4</b>
1.1. Наименование системы . . . . .	4
1.2. Основания для проведения работ . . . . .	4
1.3. Наименование заказчика и разработчика . . . . .	4
1.4. Плановые сроки начала и окончания работы . . . . .	4
1.5. Порядок оформления и предъявления заказчику результатов работ . . . . .	5
1.6. Список использованных терминов и сокращений . . . . .	5
<b>2. Назначение и цели создания системы</b>	<b>6</b>
2.1. Назначение системы . . . . .	6
2.2. Цель создания системы . . . . .	6
<b>3. Характеристика объектов автоматизации</b>	<b>7</b>
<b>4. Требования к системе</b>	<b>7</b>
4.1. Требования к системе в целом . . . . .	7
4.2. Требования к функциям, выполняемым системой . . . . .	8
4.3. Требования к видам обеспечения . . . . .	11
4.3.1. Требования к математическому обеспечению . . . . .	11
4.3.2. Требования к информационному обеспечению системы . . . . .	11
4.3.3. Требования к лингвистическому обеспечению системы	13
4.3.4. Требования к программному обеспечению системы . . . . .	14
4.3.5. Требования к техническому обеспечению системы . . . . .	14
4.3.6. Требования к метрологическому обеспечению . . . . .	15
4.3.7. Требования к организационному обеспечению . . . . .	15
4.3.8. Требования к методическому обеспечению . . . . .	16

4.3.9. Требования к лицензионной чистоте . . . . .	16
<b>5. Состав и содержание работ по созданию системы</b>	<b>16</b>
5.1. Общие положения . . . . .	16
5.2. Backlog проекта . . . . .	17
<b>6. Порядок контроля и приёмки системы</b>	<b>19</b>
<b>7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие</b>	<b>20</b>
7.1. Необходимые артефакты системы и документация . . . . .	20
7.2. Подготовка к запуску . . . . .	21
<b>8. Требования к документированию</b>	<b>21</b>
8.1. Общие сведения . . . . .	21
8.2. Перечень необходимых документов . . . . .	22
<b>Источники разработки</b>	<b>22</b>
<b>Приложение 1. Диаграмма сущность-связь (модель данных)</b>	<b>25</b>
<b>Приложение 2. UML-диаграмма прецедентов</b>	<b>26</b>

## **1. Общие сведения**

### **1.1. Наименование системы**

Полное наименование системы - информационная система управления учебным процессом с использованием гибкой методологии с элементами игрофикации

Краткое наименование системы - Agilefication

### **1.2. Основания для проведения работ**

Исследование в рамках дипломной работы по теме “Разработка комплексного ИТ-решения для применения методов игрофикации и гибкой методологии управления учебным процессом в образовательной организации”

### **1.3. Наименование заказчика и разработчика**

Заказчик - РГПУ им. А. И. Герцена, кафедра компьютерных технологий и электронного обучения

Разработчик - студент 4 курса, направления 09.03.01 “Информатика и вычислительная техника (профиль: разработка программного обеспечения)” Калинина Арина Алексеевна

### **1.4. Плановые сроки начала и окончания работы**

Начало работ - 01.09.2018 г.

Окончание работ - 30.05.2019 г.

## **1.5. Порядок оформления и предъявления заказчику результатов работ**

Система передаётся Заказчику в виде пакета исходных текстов с соответствующей сопроводительной документацией:

- руководство пользователя
- руководство системного администратора
- протокол сборки и автоматического тестирования (с описанием REST API)

Для демонстрации система может быть запущена на вычислительных ресурсах Заказчика или на независимых серверах, размещенных в облачной инфраструктуре

## **1.6. Список использованных терминов и сокращений**

- ИТ - информационная технология
- ИС - информационная система
- Git - программное обеспечение для контроля версий, разработанное Линусом Торвальдсом
- UX - User eXperience - опыт пользовательского взаимодействия

## **2. Назначение и цели создания системы**

### **2.1. Назначение системы**

Управление учебным процессом с использованием гибкой методологии с элементами игрофикации

### **2.2. Цель создания системы**

Система создаётся с целью реализации вариативных гибких методологий управления учебным процессом с использованием элементов игрофикации для поддержания мотивации, выборе оптимальной стратегии взаимодействия с учащимися, доставки образовательного содержания (с использованием технологий системы контроля версий Git).

Система может использоваться как на уровне одной дисциплины или её фрагмента (с отслеживанием достижения внутренних целей и компетенций), так и в масштабах всего образовательного процесса в университете.

Система может быть использована студентами для планирования своей деятельности по самообучению и развитию, совместно с преподавателями (в этом случае используется постановка задач со стороны преподавателя, но у студента остаётся значительная свобода в их реализации), так и полностью со стороны преподавателя с контролем выполнения промежуточных этапов (спринтов).

Информация об активностях и ошибках агрегируется в сводные аналитические отчёты, которые могут быть использованы для обнаружения узких мест и улучшения процесса.

### **3. Характеристика объектов автоматизации**

Объектом автоматизации является учебный процесс, который включает в себя виды учебной активности, систему целей и развиваемых компетенций, механизмы мотивации.

Для этапа учебного процесса (который отображается в спринт в гибкой методологии управления) характерно наличие набора достижимых целей (с указанием критериев достижимости), сроки, список участников (индивидуальных и групповых), связанные объекты в системе контроля версий.

Управление мотивацией осуществляется через метафору игрового персонажа, который развивается по шкале опыта и изменяется по шкале доверия, может изменять свои характеристики и воздействовать на других персонажей или объекты системы (задачи, спринты и др.).

Игровые персонажи могут быть персонализированы (создание аватара) и получают знаки отличия по мере развития характеристик.

### **4. Требования к системе**

#### **4.1. Требования к системе в целом**

Система разрабатывается для использования в веб-версии или через мобильное приложение и должна соответствовать общим требованиям эргономики и удобства использования для возрастной группы 17-30 лет (в соответствии с определениями из ГОСТ Р ИСО 9241-210-2016 (Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем)). Система

должна предусматривать корректное отображение оповещений об ошибочных ситуациях, без аварийного завершения работы, достаточное для выполнения базовых операций быстроедействий при пиковых нагрузках до 3000 одновременных пользователей в минуту. Интерфейс системы должен быть разработан с учётом требований адаптивности под различные разрешения, исключать графические элементы и разметку с абсолютными размерами.

Мобильное приложение должно использовать собственные элементы оформления операционной системы и быть доступно на двух основных платформах iOS и Android (через магазин приложений).

### **4.2. Требования к функциям, выполняемым системой**

Система должна реализовывать следующие категории функций:

- Административный интерфейс для управления сущностями системы;
- Интерфейс управления проектами с ориентировкой на гибкую методологию (scrum, канбан);
- Интерфейс пользователя (профиль игрового персонажа, регистрация фактов, текущая активность, внутриигровые действия), взаимодействие с канбаном

Административный интерфейс реализует управление (создание, изменение, удаление) следующих типов объектов:

- язык системы (код, название)
- список ролей пользователей (название)
- шаблон этапа канбана (цвет текста, цвет фона, варианты локализации названия, признак начальный-конечный этап)



- шаблон канбана (состав и расположение этапов, разрешения ролей по матрице разрешений)
- характеристики игрового персонажа (название, максимальное значение, перевернутая шкала, факторы временного изменения, пороговые значения и соответствующие визуальные образы)
- типологию пользователей с точки зрения отношения к игровому процессу и совместной деятельности (название)
- виды воздействий и их влияние на характеристики персонажей (название, изменения характеристик)
- классы игровых персонажей (название, связанный тип, внутриигровая легенда, аватар, коэффициент по характеристикам: при получении при реализации цели, при внешнем воздействии, при оказании воздействия, список дополнительных возможностей и их влияние на внешние объекты и характеристики персонажа)
- уровни опыта персонажа (пороговое значение, графическое изображение)
- управление метками видов деятельности (название)
- настройки интеграции с Mattermost, Gitlab, Badgr, внешними сервисами

Интерфейс управления включает в себя функции:

- управление группами (создание, изменение, удаление) для объединения отдельных пользователей или других групп
- типы фактов (и их влияние на характеристики персонажа, опыт и карму)
- управление досками (создание из шаблона, преобразование в эталонную доску, предоставление доступа пользователям и группам с указанием роли)
- управление проектами и их итерациями (создание, удаление,

изменение, формирование набора спринтов с указанием их продолжительности и связанной эталонной доски с отметкой типа использования - индивидуальное клонирование доски или общая групповая), назначение списков доступа (пользователи и группы)

- управление целями (создание, удаление, изменение, в характеристиках - пороговое значение, связанные подцели), создание связей с подцелями и их весом, обязательностью или дополнительностью, итерациями проектов, спринтами, задачами, внешними подтверждениями
- регистрация фактов и их подтверждение/опровержение

### Интерфейс пользователя:

- просмотр текущей информации по персонажу (характеристики, опыт, карма) с включением настроенных визуальных изображений
- регистрация и подтверждение/опровержение фактов
- переход в канбан для управления задачами (перемещение между этапами, внесение корректировок, удаление - в соответствии с разрешениями на конкретную роль для конкретного объекта)
- оказание воздействий на других персонажей или объекты (задачи, спринты, проекты) с использованием возможностей своего персонажа
- получение информации о последних изменениях характеристик персонажа и эффекте от выполненных воздействий
- получение информации о прогрессе в достижении актуальных целей (с возможностью маркировки целей как избранных и их просмотра на отдельной диаграмме) - отображение взаимосвязанных целей осуществляется в виде направленного графа

В описании карточки задачи могут быть указаны:

- название
- детальное описание (необязательно)
- ссылка на изображение (необязательно)
- срок выполнения (необязательно)
- контроль срока выполнения (влияет на показатели персонажа при нарушении сроков, необязательно)
- метки по видам деятельности - для агрегации в оценку порога достижения цели
- постановщик задачи (пользователь)
- ответственный (или несколько ответственных)
- влияние на характеристики (опыт, изменение иных характеристик персонажа)

### **4.3. Требования к видам обеспечения**

#### **4.3.1. Требования к математическому обеспечению**

Для передачи конфиденциальной и чувствительной информации пользователей должны использоваться алгоритмы шифрования на канале связи, сертифицированные для использования в Российской Федерации.

#### **4.3.2. Требования к информационному обеспечению системы**

Хранение информации обо всех видах действий должно осуществляться в реляционной СУБД с поддержкой транзакций, либо в документной

базе данных (в тех случаях, когда необходимо сохранить объекты сложной структуры).

Доступ к данным и управлению ими осуществляется через защищенный протокол связи с обязательной предварительной аутентификацией и авторизацией по протоколу OpenID Connect 2.0 (с внешней системой проверки подлинности).

Для передачи данных должен быть использован текстовый протокол, совместимый с HTTP, желательно использование соглашений REST.

Система осуществляет взаимодействие с другими сервисами с использованием доверенных механизмов авторизации (токен доступа или иной защищенный механизм) и может выполнять действия в сторонних системах от имени авторизованного пользователя.

Мобильное приложение и веб-версия системы могут сохранять данные внутри среды выполнения (локальном хранилище) в следующих целях:

- сохранение авторизации пользователя между запусками приложения
- обеспечение оффлайн-доступа к критически важной информации
- кэширование данных для уменьшения нагрузки на канал связи и серверную реализацию

Серверная сторона взаимодействует с программным обеспечением хранения данных только в пределах защищенного периметра сети без отправки данных через публичные сети, либо располагаться на одном физическом или виртуальном сервере с программным обеспечением хранения данных.

Учётные данные пользователя ни при каких обстоятельствах не должны без шифрования передаваться в систему или сохраняться в оперативной или долговременной памяти. Серверная сторона не сохраняет промежуточные состояния сеанса пользователя никаких

иным способом, кроме записи в долговременное разделяемое хранилище (базу данных).

Обязательно требуется автоматическое резервное копирование всей информации (баз данных, загруженных артефактов и др.) на расположение, отличное от сервера развертывания системы.

При возникновении аварийных ситуаций должна быть предусмотрена и описана процедура восстановления из резервной копии и повторного запуска системы (или сборки её из исходных текстов в виде docker-контейнеров)

### **4.3.3. Требования к лингвистическому обеспечению системы**

Разработка системы должна быть осуществлена с использованием языка программирования высокого уровня, который позволяет создавать приложения для всех трёх платформ - серверная сторона, веб-версия и мобильное приложения. Компилятор с языка программирования должен быть доступен свободно, синтаксис языка и основных библиотек специфицирован и поддерживаться крупной компанией или сообществом.

Всё программное обеспечения для организации взаимодействия должно поддерживать хранение строковых значений с использованием кодировки UTF-8, интерфейс системы поддерживать многоязычность (на этапе запуска - русский и английский язык).

#### **4.3.4. Требования к программному обеспечению системы**

Компоненты системы должны быть запущены в виде docker-контейнеров и функционировать независимо от операционной системы основной машины. Сборка система в контейнере должна включать в себя выполнение автоматических тестов, подготовку необходимых артефактов (включая документацию) и выполнение обновления на сервере публикации (в случае успешного выполнения всех предыдущих этапов). Схема базы данных, UML-диаграммы и иная документация должна вручную или автоматически синхронизироваться с содержанием модели данных и серверной реализации API.

#### **4.3.5. Требования к техническому обеспечению системы**

Требования к аппаратному обеспечению физических или виртуальных серверов определяется требованиями технологии разработки для запуска виртуальной машины (серверная реализация - не менее 1 Гб ОЗУ, 2.4GHz 4 виртуальных ядра, для веб-клиента - браузеры не старше 1 года, для мобильного приложения - версия Android не ниже 4.4 или iOS не ниже 9).

Требования к базе данных определяются общими требованиями к запуску СУБД на 200 одновременных подключений (4 Гб ОЗУ, 2.4GHz 4 виртуальных ядра)

Доставка мобильных приложений осуществляется посредством Google Play или AppStore с использованием учётной записи разработчика, предоставленной Заказчиком. Доставка веб-версии - любым веб-

сервером, поскольку требуется передача нескольких статических файлов. Серверная реализация включает в себя API (один из разрабатываемых компонентов системы) и СУБД, опубликованную под свободной лицензией для исключения юридических претензий при реальном использовании с большим количеством клиентов и возможностью потенциальной коммерциализации.

Для масштабирования должны быть предусмотрены механизмы запуска кластера сервисов (базы данных, API, веб-сервера) для горизонтального масштабирования с пересылкой трафика на один из серверов кластера (с контролем доступности).

Серверы БД и сервер API должны быть объединены защищенной локальной сетью со скоростью не менее 100 МБит/с.

### **4.3.6. Требования к метрологическому обеспечению**

Требования к метрологическому обеспечению не предъявляются

### **4.3.7. Требования к организационному обеспечению**

Основными пользователями системы являются штатные и внештатные преподаватели университета, студенты, а также независимые исследователи, заинтересованные в использовании гибкой методологии управления для реализации образовательных программ. Для работы с системой требуется знание персонального компьютера и интернет на уровне пользователя. Для всех потенциально опасных действий в системе должна быть предусмотрена возможность уведомления и подтверждения.

#### **4.3.8. Требования к методическому обеспечению**

В составе проекта должна быть приведён пример методики организации изучения одной темы курса “Программирование” с использованием всех разработанных возможностей - созданию игрового мира (на основе легенды любого из фантастических миров), определении системы целей и спринтов, определение и описания сценариев взаимодействия с системой со стороны администратора, преподавателя и учащегося.

#### **4.3.9. Требования к лицензионной чистоте**

Используемые компоненты, библиотеки, среды разработки и языки программирования должны быть доступны по свободной лицензии. Не должны использоваться коммерческие библиотеки или фрагменты иных систем, лицензированное с запретом использования в составе других систем или нарушающие патентные ограничения или требования по безопасности, согласно регламентирующим документам, действующим на территории РФ.

### **5. Состав и содержание работ по созданию системы**

#### **5.1. Общие положения**

Разработка системы производится с использованием методологии scrum, с разбиением проекта на спринты, продолжительностью 1



месяц. Это связано с непрерывным уточнением концепции проекта по мере выполнения анализа опыта исследований в области применения гибкой методологии и игрофикации в образовании.

По результатам каждого этапа (спринта) разрабатывается система автоматических тестов для исключения регрессий при добавлении дополнительных функций

Должен быть настроен автоматический конвейер сборки и тестирования, который выполняет все автоматические проверки кода, компилирует необходимые артефакты системы (контейнеры) и документации.

Для управления разработкой должна использоваться непосредственно сама система, поскольку канбан реализуется на начальных этапах, это позволяет выполнить дополнительное тестирование функциональности и внести корректировки в интерфейс с точки зрения удобства использования

Планирование спринта выполняется 3 числа каждого месяца (или в первый рабочий день после этого, если дата приходится на выходные) с разделением задач высокого уровня на конкретные задачи. Оценка прогресса должна выполняться каждый понедельник (с изучением диаграммы сгорания) с внесением корректировок в план ближайшей недели (при необходимости)

## 5.2. Backlog проекта

Перечень задач в backlog проекта:

- выбор технологий разработки для API, веб и мобильной платформы

- изучение и тестирование возможностей создания автоматической документации (REST API, диаграммы сущность-связь, иная документация)
- изучение возможностей и разработка минимальных тестов для API, инструментального тестирования веб- и мобильных приложений
- анализ психологического профиля и типологии игроков
- реализация веб-версии канбана для отображения и перемещения карточек
- пакетный импорт и экспорт задач из этапа
- разработка конфигурации сборки и развертывания системы с использованием docker
- разработка модели управления права доступа в соответствии с ролями
- проектирование модели игрового персонажа, его характеристик
- проектирование классов игровых персонажей и модели их взаимодействия между собой и с объектами системы
- разработка модели ступеней по уровням опыта и по пороговым значениям характеристик
- проектирование модели внешних событий (фактов) и их подтверждения
- разработка видов активности по задачам и модели контроля завершения этапа по сочетанию задач и их влияния
- разработка модели доверия (кармы)
- разработка моделей целей, их взаимосвязи
- реализация административного интерфейса для управления (API + Web-приложение)
  - ролями
  - языками

- шаблонами этапов
  - шаблонами досок
  - проектами и их итерациями
  - целями и их взаимосвязями
- проектирование пользовательского интерфейса и их веб-приложения
- проектирование пользовательского интерфейса и их для мобильного приложения
- разработка интерфейса веб-приложения для преподавателя и слушателя
- создание программной реализации интерфейса веб-приложения
- разработка интерфейса мобильного приложения для преподавателя и слушателя
- создание программной реализации интерфейса мобильного приложения
- интерфейсы администратора для просмотра журнала действий над объектами системы

## 6. Порядок контроля и приёмки системы

Приёмка системы осуществляется по результатам выполнения серии автоматических тестов, прохождения сценарного тестирования согласно разработанному и утвержденному Исполнителем и Заказчиком плану, по результатам пробной эксплуатации на небольшой контрольной группе.

Сценарий тестирования включает в себя проверку административного интерфейса, создание и управление объектами из интерфейса

преподавателя, проверка доступа к материалам и возможности взаимодействия с объектами системы со стороны слушателя. Набор тестов полностью определяется п.4.2. настоящего документа.

По итогам прохождения тестов обоими сторонами подписывается акт о принятии системы в опытную эксплуатацию.

## **7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие**

### **7.1. Необходимые артефакты системы и документация**

Система передаётся в виде набора docker-контейнеров, предназначенных для запуска на Docker Engine не менее 18.03. Обязательным является наличие shell-сценария запуска необходимых контейнеров и создания взаимосвязей и постоянного хранилища (база данных, API, сервер для доставки веб-версии). Для автоматической доставки обновлений рекомендуется также включить конфигурацию публикации в кластер Kubernetes, которая минимально должна включать в себя следующие компоненты:

- описание развертывания контейнеров и их среды выполнения (deployment) с обязательным включением liveness и readiness-проверок
- описание сервисов и их взаимосвязи с pod (service)
- описание запросов дисковых разделов для постоянного хранения информации
- описание ingress-правил для привязки веб-адреса к сервисам

## **7.2. Подготовка к запуску**

Для запуска системы требуется подготовить тестовый набор данных, который должен быть загружен в базу данных на этапе пилотного запуска информационной системы. Перечень необходимой информации:

- языки системы (русский, английский)
- типовые шаблоны этапов (сделать, в процессе, на проверке, завершено)
- типовые роли (преподаватель, студент, куратор)
- типовой шаблон канбана и модель разрешений для различных ролей
- загруженная типология игроков (согласно User Types Hexad)
- информация о классах персонажей (на примере мифологии миров Братства Кольца)
- типовые характеристики персонажа (здоровье, интеллект, общительность)
- пороговые значения для 3 уровней опыта (0, 100, 500)

## **8. Требования к документированию**

### **8.1. Общие сведения**

Отчётная документация должна передаваться Заказчику в бумажном и электронном виде на русском языке. Вспомогательная документация (не указанная в качестве непосредственного результата работ) передаётся только в электронном виде. Техническая и эксплуатационная документация на Систему (далее - документы на Систему) должна удовлетворять требованиям комплекса стандартов и руководящих

документов на автоматизированные системы: ГОСТ 34.003-90 - в части терминологии; ГОСТ 34.201-89, ГОСТ 19.101-77-82, 19.103-77 - в части наименования и обозначения документов; ГОСТ 34.601-90 - в части определения стадий и этапов работ; ГОСТ 34.602-89 - в части состава, содержания и правил оформления документов «Техническое задание», «Частное техническое задание». ГОСТ 34.603 -92 - в части определения видов испытаний; РД 50-34.698-90 - в части структуры и содержания документов.

### **8.2. Перечень необходимых документов**

Документация на систему должна обязательно включать в себя

- инструкцию пользователя
- инструкцию администратора
- руководство по установке, резервному копированию и восстановлению после сбоев
- описание моделей данных и REST API, UML-диаграммы
- протокол сборки (журналы сборочных этапов, прохождение автоматических тестов)

### **Источники разработки**

При разработке настоящего технического задания использовались следующие документы:

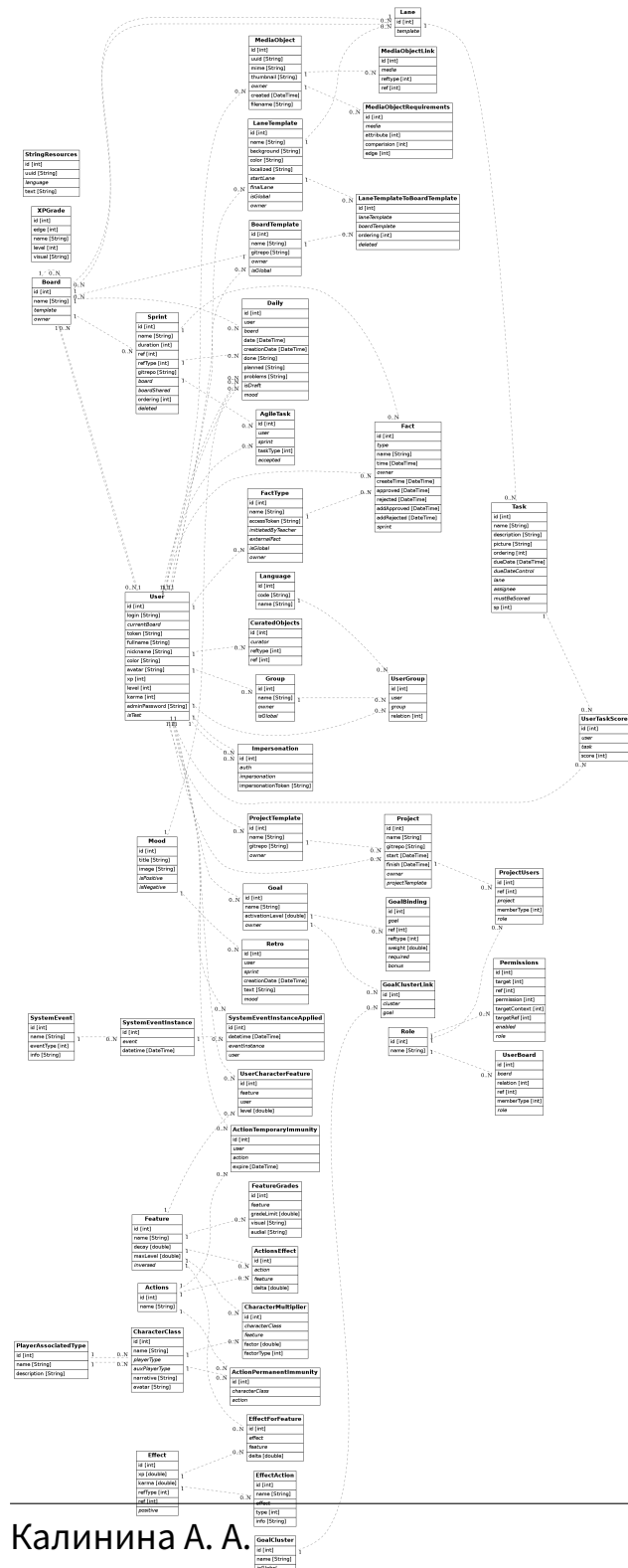
- ГОСТ 19.004-80 «Единая система программной документации. Термины и определения»;

- ГОСТ 19.101-77 «Единая система программной документации. Виды программ и программных документов»;
- ГОСТ 19.102-77 «Единая система программной документации. Стадии разработки»;
- ГОСТ 19.201-78-82 «Единая система программной документации. Техническое задание. Требования к содержанию и оформлению»;
- ГОСТ 34.003-90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения»;
- ГОСТ 34.201-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем»;
- ГОСТ 34.601-90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Стадии создания»;
- ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»;
- ГОСТ 34.603-92 «Информационная технология. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем»;
- РД 50-34.698-90 «Автоматизированные системы. Требования к содержанию документов»;
- ГОСТ Р ИСО 9241-110-2016 Эргономика взаимодействия человек-система. Часть 110. Принципы организации диалога;
- ГОСТ Р ИСО 9241-210-2016 Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем.





# Приложение 1. Диаграмма сущность-связь (модель данных)



## Приложение 2. UML-диаграмма прецедентов

