

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Инженерно-экономический факультет

Кафедра экономической информатики

Дисциплина: Средства и технологии анализа и разработки
информационных систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему:

«Портал для любителей «Что? Где? Когда?»»»

Выполнила:

студент гр.673601
Кавзович Анастасия Викторовна

Проверил:

асс. кафедры
Хомяков Павел Васильевич

Минск 2019

СОДЕРЖАНИЕ

Введение.....	5
1 Описание игры «Что? Где? Когда?».....	6
2 Описание процесса проведения спортивной версии игры «Что? Где? Когда?»	9
3 Спецификация вариантов использования системы	13
4 Информационная модель системы и её описание	15
5 Обоснование выбора компонентов и технологий для реализации курсового проекта	17
6 Модели представления системы и их описание.....	21
7 Руководство по развертыванию системы	22
8 Результаты тестирования системы.....	24
Заключение	26
Список используемых источников.....	28
Приложение А (обязательное) Листинг основных элементов программы.....	29
Приложение Б (обязательное) Функциональная модель системы.....	30
Приложение В (обязательное) Листинг скрипта генерации базы данных	34
Приложение Г (обязательное) Диаграмма классов	37

ВВЕДЕНИЕ

«Что? Где? Когда?» — это популярная интеллектуальная игра, участником которой может стать любой желающий. Для победы не требуются специальных знаний. Нужен лишь творческий подход, логика, фантазия, интуиция и командная работа.

Интеллект — понятие многогранное, которое включает в себя много составляющих. У каждого человека эти составляющие развиты в разной мере. Преимущество такого рода игр как «Что? Где? Когда?» (далее ЧГК) в том, что здесь могут найти себя люди с разным типом мышления. ЧГК — это не просто игра, а полезный для жизни и бизнеса тренинг. Ведь во время игры расширяется кругозор участников, повышается интеллектуальная культура, формируются навыки эффективного командного взаимодействия, проводится пропаганда научных знаний и развивается интерес к научной деятельности.

Клубы ЧГК существуют не только в странах бывшего СССР, но и во многих странах мира (Германии, Израиле, США, Великобритании, Канаде, Финляндии, Чехии, Нидерландах, Польше, Кипре, Швейцарии, Румынии). «Что? Где? Когда?» очень быстро стала настолько популярной, что кроме традиционной игры, появилась её спортивная версия, в которой одновременно могут принимать участие большее количество игроков.

С появлением Всемирной паутины появилась возможность предоставить доступ к игре еще большему количеству желающих и создать новый формат игры, позволяющий играть в нее детям и взрослым, людям разных профессий и разного уровня эрудиции. Целью данной работы является содействие повышению интеллектуального и культурного уровня пользователей интернета, популяризация форм интеллектуального досуга, а также популяризация и распространение самой игры посредством создания интернет-портала для любителей «Что? Где? Когда?».

Поставленная цель потребовала решения следующих задач:

- исследовать предметную область;
- спроектировать модель системы в целом;
- разработать структуры хранимых данных;
- разработать диаграмму классов на языке UML;
- закодировать систему и провести тестирование;
- создать документацию.

Ключевые слова: ПОПУЛЯРИЗАЦИЯ, ИНТЕЛЛЕКТ, ИНТЕРНЕТ-ПОРТАЛ, «ЧТО? ГДЕ? КОГДА?».

1 ОПИСАНИЕ ИГРЫ «ЧТО? ГДЕ? КОГДА?»

«Что? Где? Когда?» — советская и российская интеллектуальная телевизионная игра, телевизионный интеллектуальный элитарный Клуб «знатоков».

Используя метод мозгового штурма, команда из шести игроков — «знатоков» — ищет в течение одной минуты правильный ответ на любой заданный вопрос. За правильный ответ очко получает команда знатоков, за неправильный же очко получают их противники — команда телезрителей. Побеждает команда, первой набравшая шесть очков.

Телеигра была создана режиссёром Владимиром Ворошиловым и редактором Наталией Стеценко в СССР. Первый выпуск передачи вышел в эфир 4 сентября 1975 года. Правила игры в первые годы существенно отличались от нынешних. «Что? Где? Когда?» очень быстро стала настолько популярной, что кроме оригинальной телевизионной игры, появилась её спортивная версия [1]. Наибольшую популярность игра получила в русскоязычной среде.

В настоящее время спортивная версия игры достаточно популярна среди различных возрастных групп. Данная версия игры имеет много отличий, например, знатоки играют не против телезрителей, а против других команд так, как в данной версии игры несколько команд могут играть одновременно.

Известны следующие разновидности игры:

- очная — все команды играют в одном зале, вопросы зачитывает один и тот же ведущий;
- синхронная — одни и те же вопросы играют одновременно (с допустимым разбросом в 3—4 дня) на различных игровых площадках (в разных городах и иногда — странах) с разными ведущими;
- строго синхронная — голос ведущего транслируется в режиме реального времени через Интернет либо допустимая задержка времени начала игры составляет не более 10 минут.

Основные правила, рекомендованные МАК для всех турниров, описаны в кодексе ЧГК [2]. Далее приведены ключевые из них:

- В команде не допускается единовременное участие в игре более чем шести человек. Разрешается наличие ограниченного числа запасных игроков в команде и замены игроков между турами. Один из игроков команды является её капитаном. Капитан ко-

манды имеет особые права (например, право обращаться в игровое жюри в перерывах между турами).

- Ответы сдаются секундантам (помощникам ведущего, часто называемым «ласточками») в письменном виде на карточках, выдаваемых команде перед началом игры.
- На обсуждение каждого вопроса выделяется 1 минута (для блица, состоящего из произвольного числа вопросов (как правило, 2—3), суммарное время обсуждения составляет также 1 минуту), после окончания которой даётся 10 секунд для записи ответа. К моменту окончания времени для записи ответа капитан (или другой игрок) обязан поднять карточку с ответом над головой, в противном случае ответ может быть не засчитан. Минута обсуждения начинается и заканчивается по сигналу ведущего; также ведущий объявляет о 50-й секунде обсуждения и об окончании времени для записи ответов.
- Основной показатель, по которому определяется место команды, — количество правильных ответов. При равенстве количества правильных ответов команды делят соответствующие места либо могут использоваться дополнительные показатели (в зависимости от регламента турнира). При равенстве количества правильных ответов у команд, претендующих на призовые места, может использоваться т. н. «перестрелка» — дополнительные вопросы, задаваемые соответствующим командам по одному до нарушения равенства.
- Если ответ не в точности совпадает с авторским, он может быть зачтён либо игровым жюри (на игре), либо апелляционным жюри (в случае подачи апелляции) в том случае, если он соответствует указанным автором критериям зачёта, является более точным, чем авторский, содержит дополнительную информацию, не меняющую его смысла, и т. д.
- Апелляционное жюри рассматривает апелляции двух типов:
 1. Требования зачёта ответа, не предусмотренного автором, но удовлетворяющего всем условиям вопроса (так называемая «дуаль»); в случае удовлетворения апелляции ответ засчитывается апеллировавшей команде, а также всем остальным командам, давшим этот или эквивалентный этому ответ).

2. Требования снятия вопроса (в случае наличия в нём любой фактической ошибки либо существенной фактической ошибки, в зависимости от регламента конкретного турнира; в случае удовлетворения апелляции вопрос исключается из числа зачётных, со всех команд снимаются очки, начисленные за его взятие).

2 ОПИСАНИЕ ПРОЦЕССА ПРОВЕДЕНИЯ СПОРТИВНОЙ ВЕРСИИ ИГРЫ «ЧТО? ГДЕ? КОГДА?»

Организация мероприятий, в том числе спортивной версии игры – увлекательный, но очень сложный, ответственный процесс, требующий временные, идейные, человеческие ресурсы. И чем грандиознее мероприятие, тем больше этих ресурсов требуется. Для проведения мероприятий на высоком уровне необходимо учесть все нюансы. С помощью нотации IDEF0 [6] формализован процесс проведения игры «Что? Где? Когда?» с точки зрения организаторов игры. На рисунке 2.1 изображен контекстный уровень функциональной модели, построенной по принципу «Как есть».

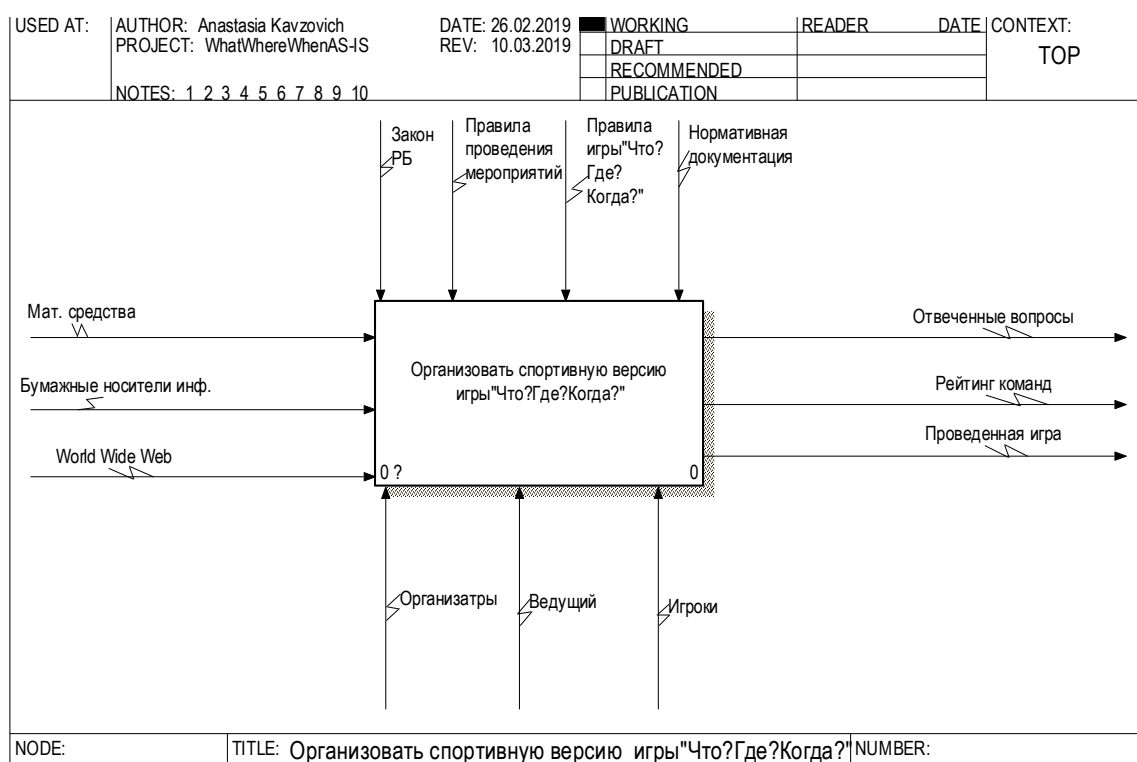


Рисунок 2.1 – Контекстный уровень функциональной модели AS-IS

Организация проведения игры определяется нормативной документацией и правилами проведения игр, правилами конкретной игры и законом страны.

Входными данными являются материальные средства, всемирная паутина и другие бумажные носители информации для поиска вопросов.

Механизмами являются организаторы, ведущий, который проводит игру и сами игроки.

На выходе имеем проведенную игру, ответенные вопросы и рейтинг команд.

Декомпозиция контекстной диаграммы представлена на рисунке 2.2. Процесс организации игры можно разделить на 4 под-процесса: подготовка игровой площадки, подготовка вопросов, непосредственно проведение игры и подведение итогов.

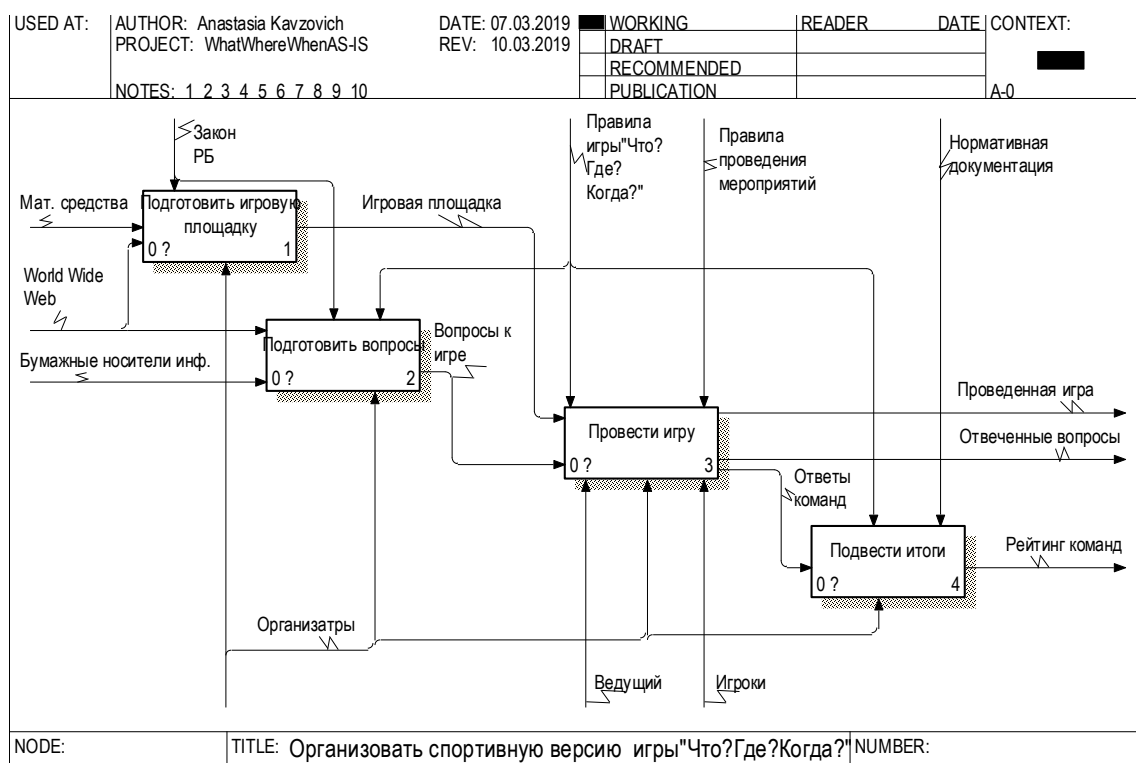


Рисунок 2.2 – Первый уровень функциональной модели AS-IS

На рисунках Б1, Б2, Б3 и Б.4 приложения Б изображены диаграммы 4 основных под-процессов.

Модель «Как есть» была проанализирована и некоторые процессы были заменены с целью их оптимизации и переноса игры в интернет. В результате была получена модель «Как должно быть». На рисунке 2.3 изображен контекстный уровень функциональной модели, построенной по принципу «Как должно быть». После реорганизации существующего бизнес-процесса изменились механизмы в модели ТО-ВЕ используются игроки, разработчики портала «Что? Где? Когда?» и администратор. Особенностью является то, что игроки смогут играть не в командах, а индивидуально, не конкретные игры, а набор вопросов.

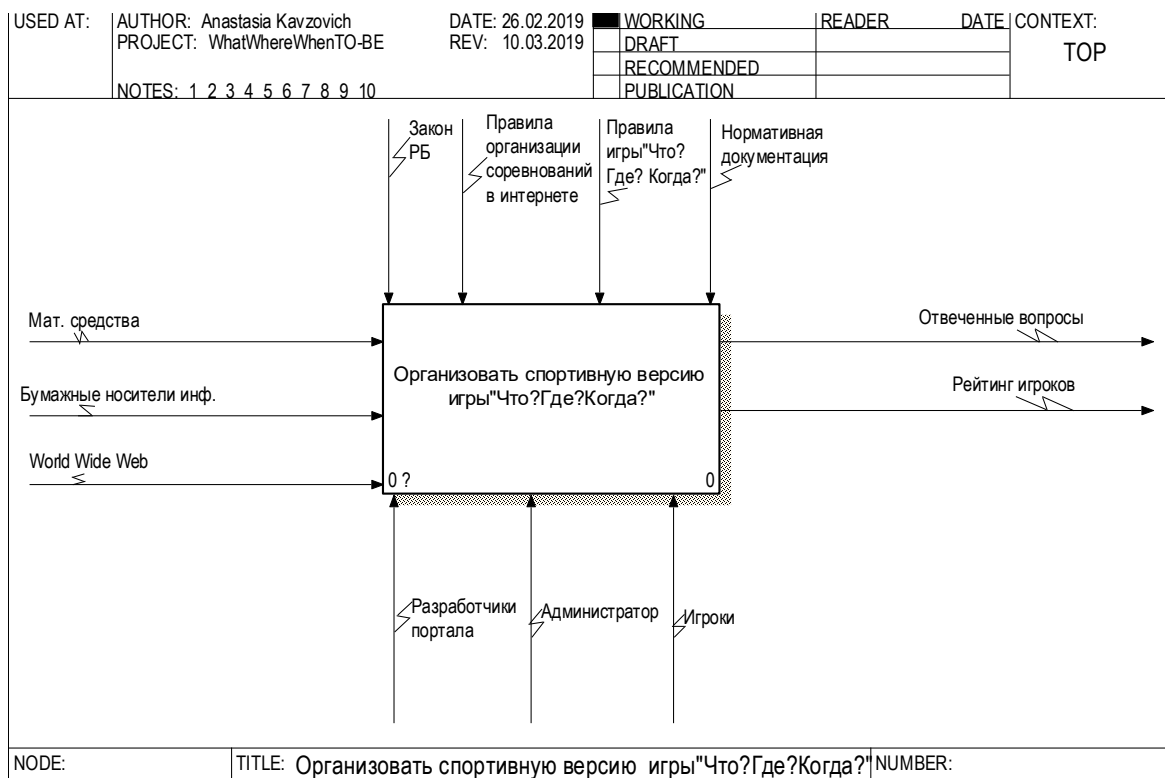


Рисунок 2.3 – Контекстный уровень функциональной модели ТО-ВЕ

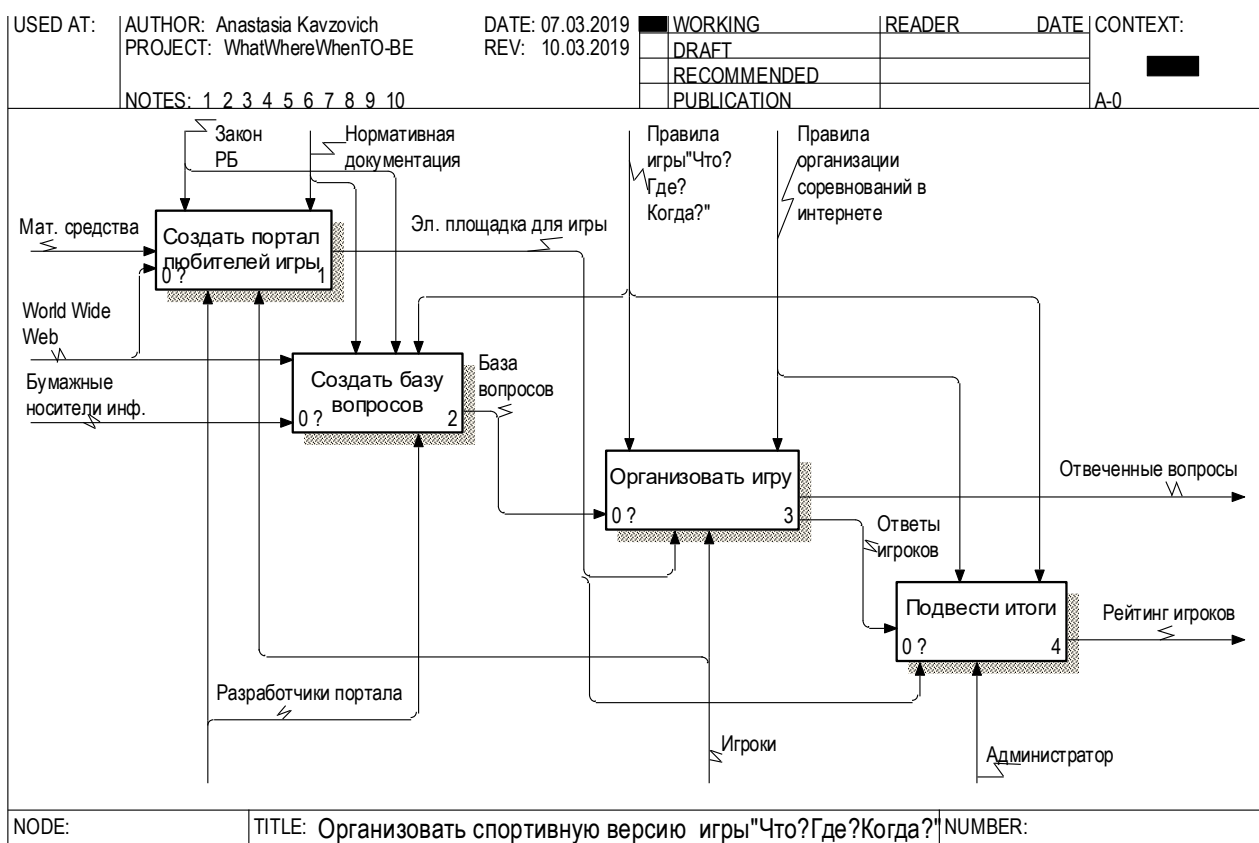


Рисунок 2.4 – Первый уровень функциональной модели ТО-ВЕ

Декомпозиция контекстной диаграммы ТО-ВЕ представлена на рисунке 2.4. Процесс организации игры в интернет-пространстве можно разделить на 4 под-процесса: создание портала любителей игры, создание базы вопросов, организации игры и подведения итогов. В приложении Б на рисунках Б.5, Б.6, Б.7, Б.8 изображены диаграммы 4 основных под-процессов модели ТО-ВЕ.

Таким образом был смоделирован бизнес-процесс организации спортивной версии игры «Что? Где? Когда?» и проведена его реорганизация для переноса игры в интернет-пространство.

3 СПЕЦИФИКАЦИЯ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ

Визуальное моделирование в UML можно представить, как некоторый процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели исходной системы к логической, а затем и к физической модели соответствующей программной системы [3]. Для достижения этих целей вначале строится модель в форме так называемой диаграммы вариантов использования (use case diagram), которая описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки. На рисунке 3.1 представлена диаграмма вариантов использования разрабатываемой системы.

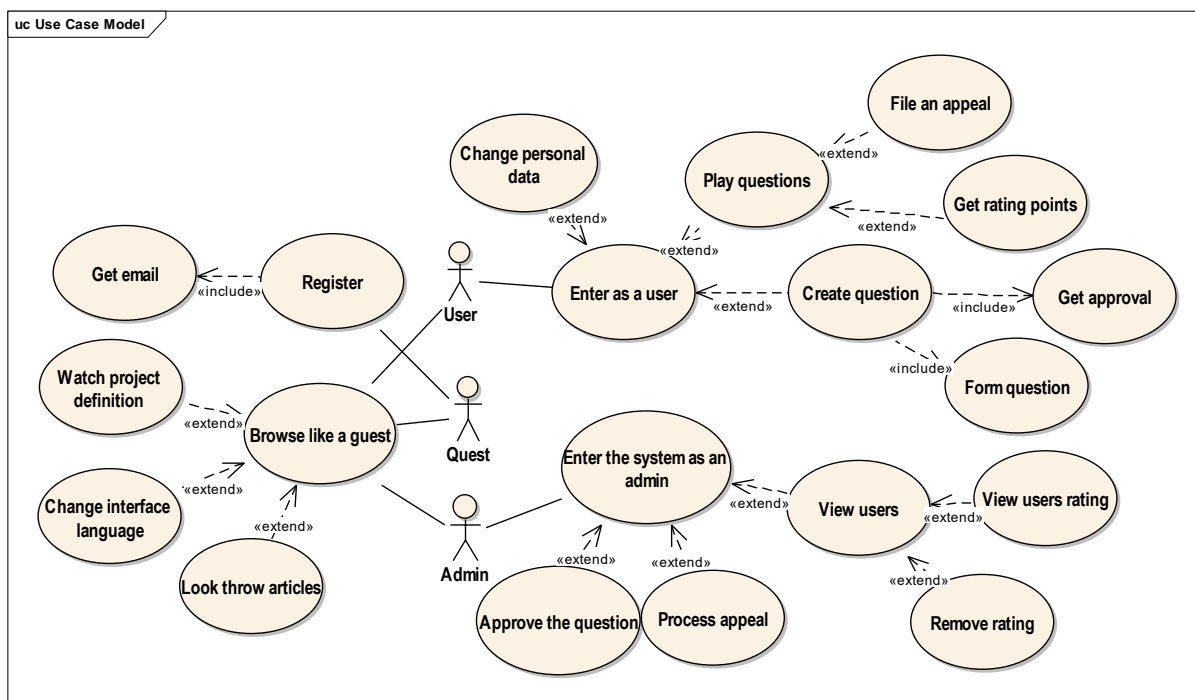


Рисунок 3.1 – Диаграмма вариантов использования

Рассматриваемая система имеет трех актеров: администратора, пользователя, гостя.

Гость – это незарегистрированный пользователь, который имеет ограниченный набор функций. Гостю доступна возможность просмотра статей, новостей и информации о проекте. Также гость всегда имеет возможность зарегистрироваться в системе и стать пользователем.

Пользователь же имеет доступ к базе вопросов и может отвечать на вопросы, создавать свои, подавать апелляции в случае необходимости, а также соревноваться с другими пользователями. У каждого пользователя есть личный кабинет, в котором отображаются его данные и рейтинг.

В системе предусмотрена роль администратора, который вправе утверждать вопросы, присланные пользователями, разрешать апелляции, начислять баллы, а также банить пользователей, нарушающих правила пользования порталом.

4 ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ И ЕЁ ОПИСАНИЕ

База данных интернет-портала строится на основе данных, добавляемых пользователями и администраторами системы, ответственность за достоверность информации, содержащейся в публикациях, несут лица, их публикующие.

С помощью интернет-портала проводится сбор и обработка информации, имеющей отношение к игре «Что? Где? Когда?». Некоторые данные, например, вопросы для игры могут являться результатами интеллектуальной деятельности пользователей портала. Для поддержания работоспособности системы потребуется постоянное обновление и добавление данных.

В результате обработки исходных данных пользователю предоставляется актуальная информация по выбранной тематике. Использование предлагаемого интернет-портала предоставит пользователям возможность интерактивной работы и удаленного доступа к необходимой информации, а также позволит получать информацию в мультимедийном виде.

Так, как данный портал предназначен для работы с большими объемами данных, поэтому необходимо создать наиболее оптимальную информационную модель системы. В связи с этим необходимо провести нормализацию базы данных [8]. Целью нормализации является исключение избыточного дублирования данных, которое является причиной аномалий, возникающих при добавлении, редактировании и удалении кортежей (строк таблицы).

Ниже описаны 3 нормальные формы (далее НФ) базы данных.

Отношение находится в 1НФ, если все его атрибуты являются простыми, все используемые домены должны содержать только скалярные значения. Не должно быть повторений строк в таблице.

Отношение находится во 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут неприводимо зависит от Первичного Ключа (ПК).

Неприводимость означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость.

Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. Проще говоря, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

В соответствии с вышеизложенными правилами была спроектирована информационная модель системы. На рисунке 4.1 изображена схема информационной модели системы, легко убедиться, что она соответствует трем нормальным формам.

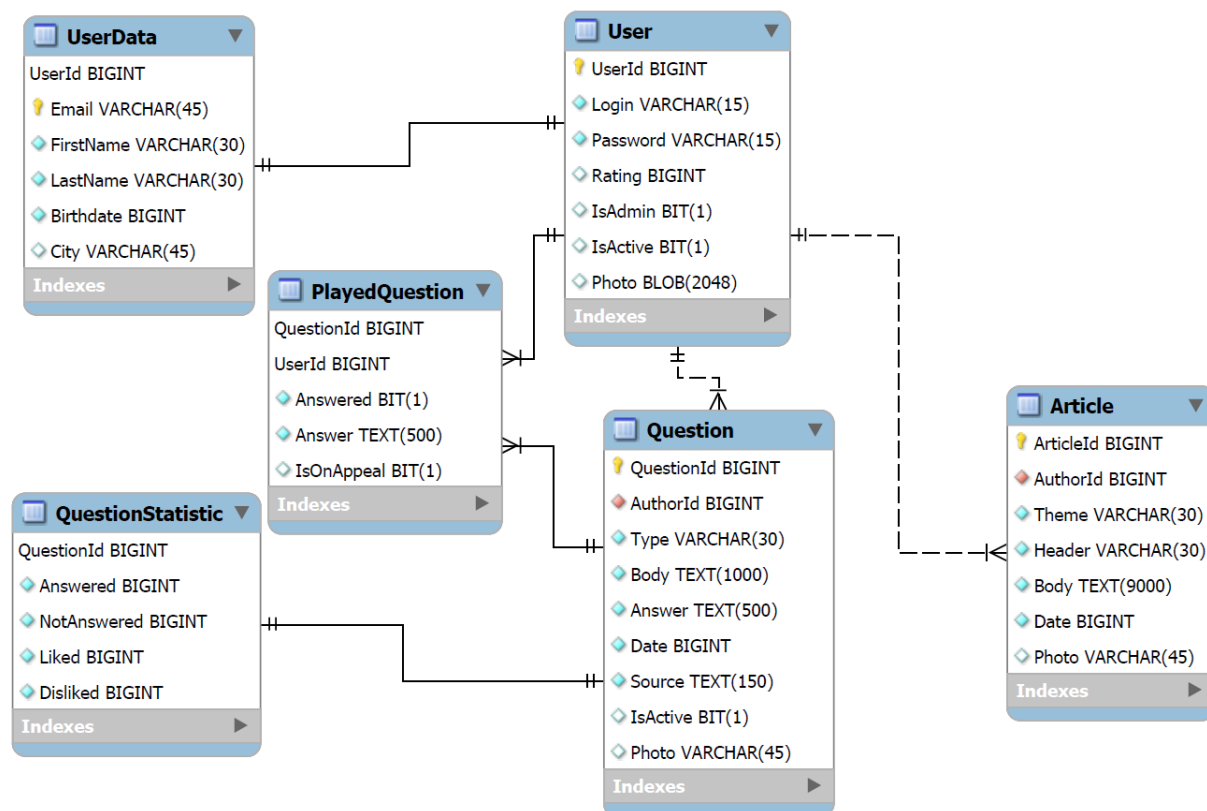


Рисунок 4.1 – Информационная модель системы

Информационная модель представлена 6 таблицами, в которых хранятся все необходимые данные информационного портала.

5 ОБОСНОВАНИЕ ВЫБОРА КОМПОНЕНТОВ И ТЕХНОЛОГИЙ ДЛЯ РЕАЛИЗАЦИИ КУРСОВОГО ПРОЕКТА

При разработке WEB приложения можно использовать одну из наиболее популярных технологий, к которым следует отнести Java Servlets, Java Server Page, PHP, ASP.NET, Node.js.

В данном курсовом проекте была использована технология Java Servlets (сервлеты), которая была разработана компанией Sun Microsystems, чтобы использовать преимущества платформы Java для решения проблем технологии CGI и API расширений сервера [5]. Технология решает проблему производительности, выполняя все запросы как потоки в одном процессе. Сервлеты не зависят от платформы поскольку выполняются внутри Java Virtual Machine (JVM), и могут легко разделять ресурсы.

Модель безопасности Java обеспечивает управление уровнем доступа, а обработка исключений делает сервлеты более надежным средством. Технология обладает широкими функциональными возможностями, и большое количество библиотек предоставляет самые разнообразные средства, необходимые в разработке.

Технология сервлетов является распространенной и может быть использована со всеми популярными WEB серверами, выполняющими функции контейнера сервлетов (Apache Tomcat, Java Web Server от Sun).

Программный интерфейс позволяет сервлетам обрабатывать запросы на низком уровне (заголовки запросов, их тип, и т.д). Это обеспечивает большую гибкость при разработке нестандартных обработчиков, например при работе с двоичным или мультимедийным содержимым.

В связи с тем, что сервлеты обрабатываются в одном процессе с помощью создания внутри него потоков, программный код сервлетов должен быть потокобезопасным. Это накладывает определенную ответственность на программиста. При этом сервлеты приобретают такое неоценимое преимущество как масштабируемость.

Таким образом, сервлеты обеспечивают компонентный, платформонезависимый метод построения WEB приложений без ограничений производительности. Они имеют широкий диапазон доступных прикладных API, позволяют использовать все преимущества Java, легко расширяются и масштабируются, поддерживаются всеми популярными WEB серверами. Все это обеспечивает использование данной технологии для разработки крупных WEB систем.

Так же использовалась технология Java Server Pages (JSP), которая является надстройкой над технологией Java Servlets, обеспечивающая более быструю и простую разработку WEB приложений с помощью использования шаблонов. Для понимания архитектуры и преимуществ JSP необходимо знать технологию Java Servlets, поскольку они тесно связаны.

Технология JSP удачно объединяет все преимущества Java платформы и шаблонный подход к построению сайтов. Благодаря этому технология получила широкое распространение как среди профессиональных коммерческих разработчиков, так и при создании открытых бесплатных проектов.

Важным шагом к расширению шаблонного подхода стали так называемые библиотеки тэгов (tag libraries). Это гибкая возможность интегрировать стандартные, сторонние, или собственные программные компоненты в страницы. Простота создания и использования привели к большой популярности библиотек тэгов.

Технология JSP не привязана к конкретной аппаратной или программной платформе благодаря работе на основе Java. То есть JSP являются отличным решением для использования в гетерогенных средах.

Основными достоинствами JSP является простота разработки, характерная для шаблонного подхода, наличие большого количества сторонних библиотек, легкость их использования, мощные и разнообразные среды разработки.

Благодаря всем вышеперечисленным факторам JSP и Java Servlets является наиболее перспективной базовой технологией разработки при создании Web-сайтов.

6 МОДЕЛИ ПРЕДСТАВЛЕНИЯ СИСТЕМЫ И ИХ ОПИСАНИЕ

Создаваемое программное обеспечение (ПО) постоянно усложняется. При работе над любыми информационными (распределенными) системами в первую очередь возникает проблема взаимопонимания программиста и заказчика уже на стадии обсуждения структуры системы. До начала кодирования программы предлагаемая концепция предусматривает решение двух предварительных задач: проанализировать поставленную перед разработчиком задачу и разработать проект будущей системы. Программа разбивается на отдельные модули, взаимодействующие между собой с помощью механизмов передачи параметров.

Унифицированный язык моделирования (Unified Modeling Language) – графический язык визуализации, специфицирования, конструирования и документирования программного обеспечения. С помощью UML можно разработать детальный план создаваемой системы, отображающий системные функции и бизнес-процессы, а также конкретные особенности реализации.

На рисунке 6.1 изображена диаграмма развертывания, которая показывает топологию системы.

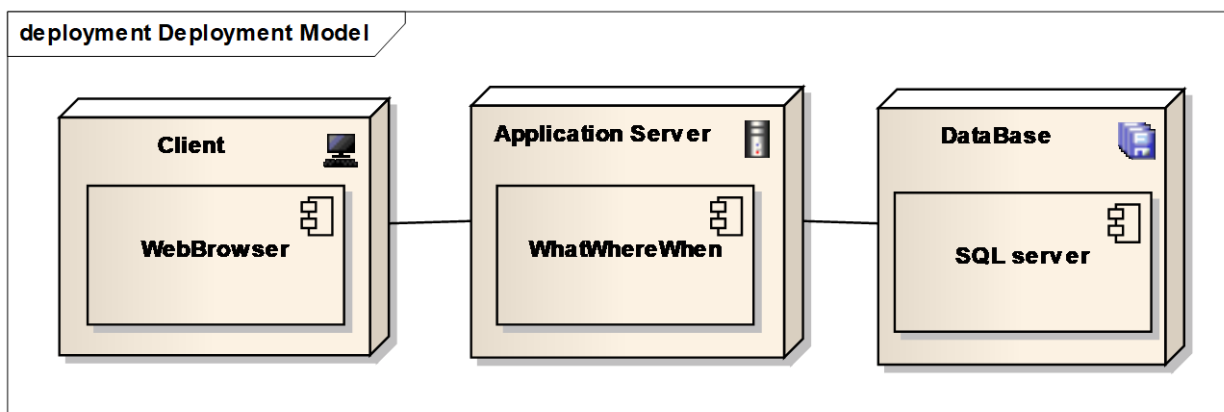


Рисунок 6.1 – Диаграмма развертывания системы

На рисунке 6.2 изображена диаграмма состояний, показывающая переходы между состояниями при создании вопроса пользователем системы.

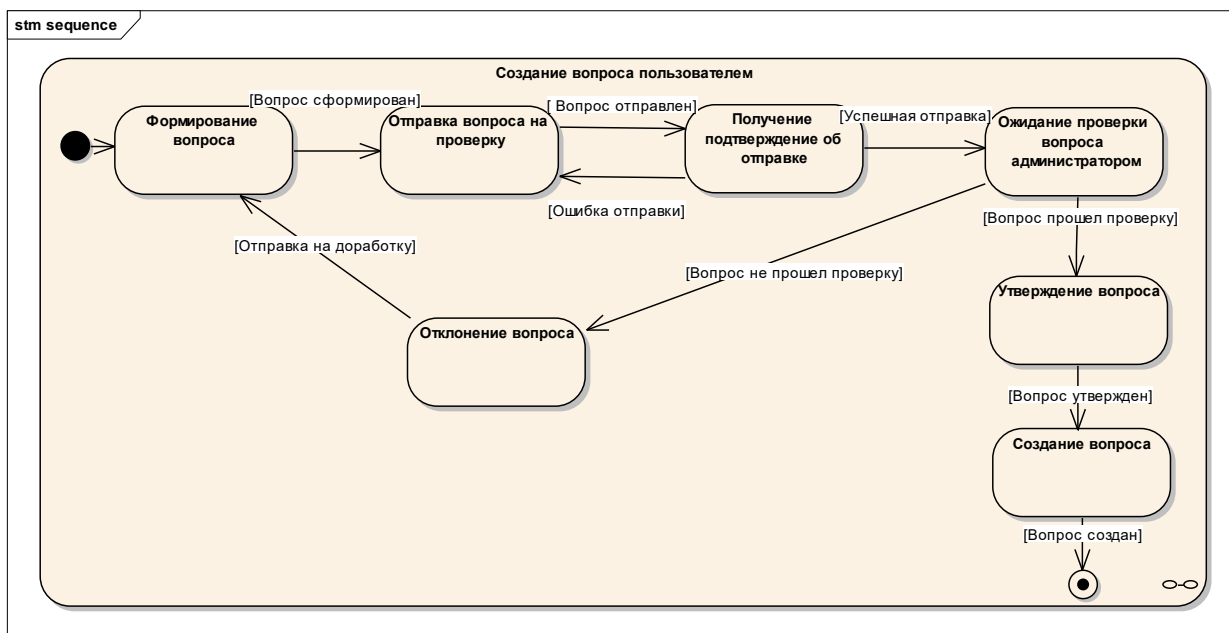


Рисунок 6.2 – Диаграмма состояний системы

На рисунке 6.3 представлена диаграмма компонентов, она показывает структуру системы.

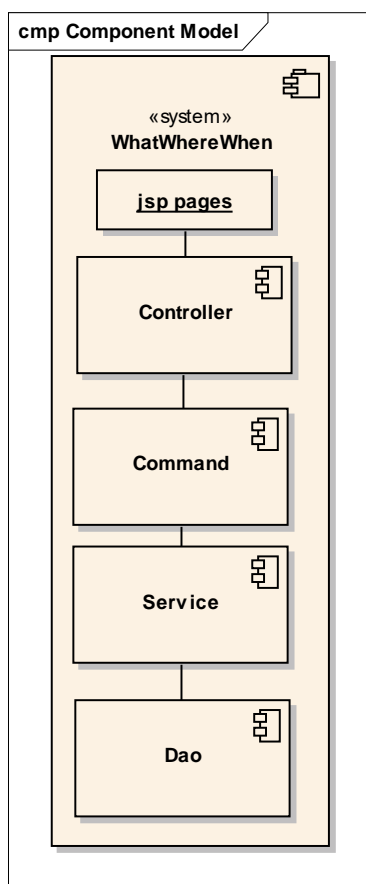


Рисунок 6.3 – Диаграмма компонентов

Взаимодействие объектов системы представлено с помощью диаграммы последовательностей, изображенной на рисунке 6.4.

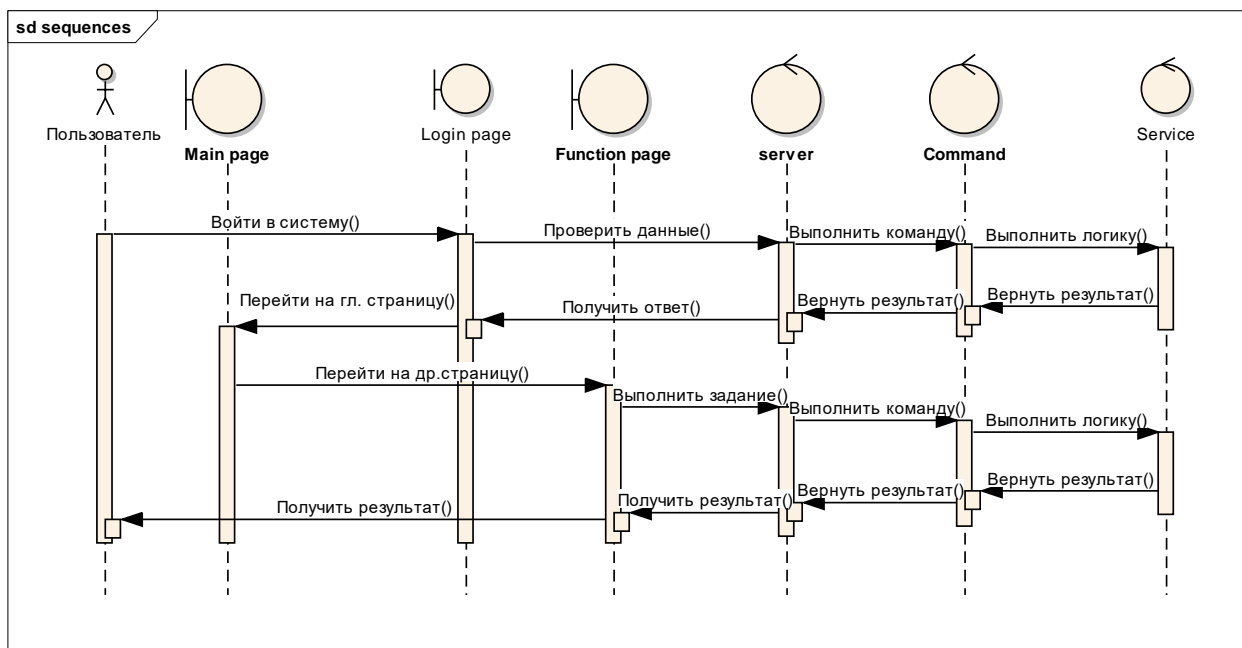


Рисунок 6.4 – Диаграмма последовательностей

На рисунке Г.1 приложения Г представлена диаграмма классов системы. Использованы следующие паттерны: поведенческий паттерн Command, порождающий паттерн Singleton и MVC(Model/View/Controller) [10].

7 РУКОВОДСТВО ПО РАЗВЕРТЫВАНИЮ СИСТЕМЫ

Для запуска Web-приложения необходимо установить веб-сервер, произвести настройки, создать базу данных с помощью скрипта и развернуть приложения.

В качестве веб-сервера использовался Apache Tomcat. Apache Tomcat – это серверное ПО для исполнения web-приложений, написанных на Java. Apache Tomcat - один из проектов организации Apache Software Foundation и является открытой реализацией таких технологий, как Java Servlet, JavaServer Pages, Java Expression Language и Java WebSocket. Он может быть использован в качестве самостоятельного веб-сервера, в качестве сервера контента в сочетании с веб-сервером Apache HTTP, а также в качестве контейнера сервлетов в некоторых серверах приложений.

Процесс установки веб-сервера описан в статье «Apache Tomcat»[9]. Когда веб-сервер установлен, а база данных сгенерирована можно приступить к развертыванию приложения. Существует два способа развертывания приложений на сервере Apache Tomcat.

Способ 1. Без использования менеджера приложений:

1. Скопировать war-файл в каталог <TOMCAT_HOME>\webapps
2. Если Tomcat был запущен, то он сам развернет приложение из war-файла через несколько секунд (появится одноименный каталог). Либо самостоятельно перезапустите Tomcat, и он развернёт war.
3. Проверьте работоспособность приложения, пройдя по ссылке <http://localhost:8080/WhatWhereWhen>.

Способ 2. С использованием менеджера приложений (Tomcat Web Application Manager):

1. Убедитесь, что Tomcat запущен.
2. Перейдите на стартовую страницу: http://IP-адрес_сервера:8080
3. Откройте менеджер web-приложений
4. Введите данные в аутентификации
5. В разделе «Deploy» нажмите кнопку возле «Select WAR file to upload» и выберите файл, который вы хотите развернуть (рисунок 7.1).
6. Нажмите кнопку «Deploy» под этой надписью.

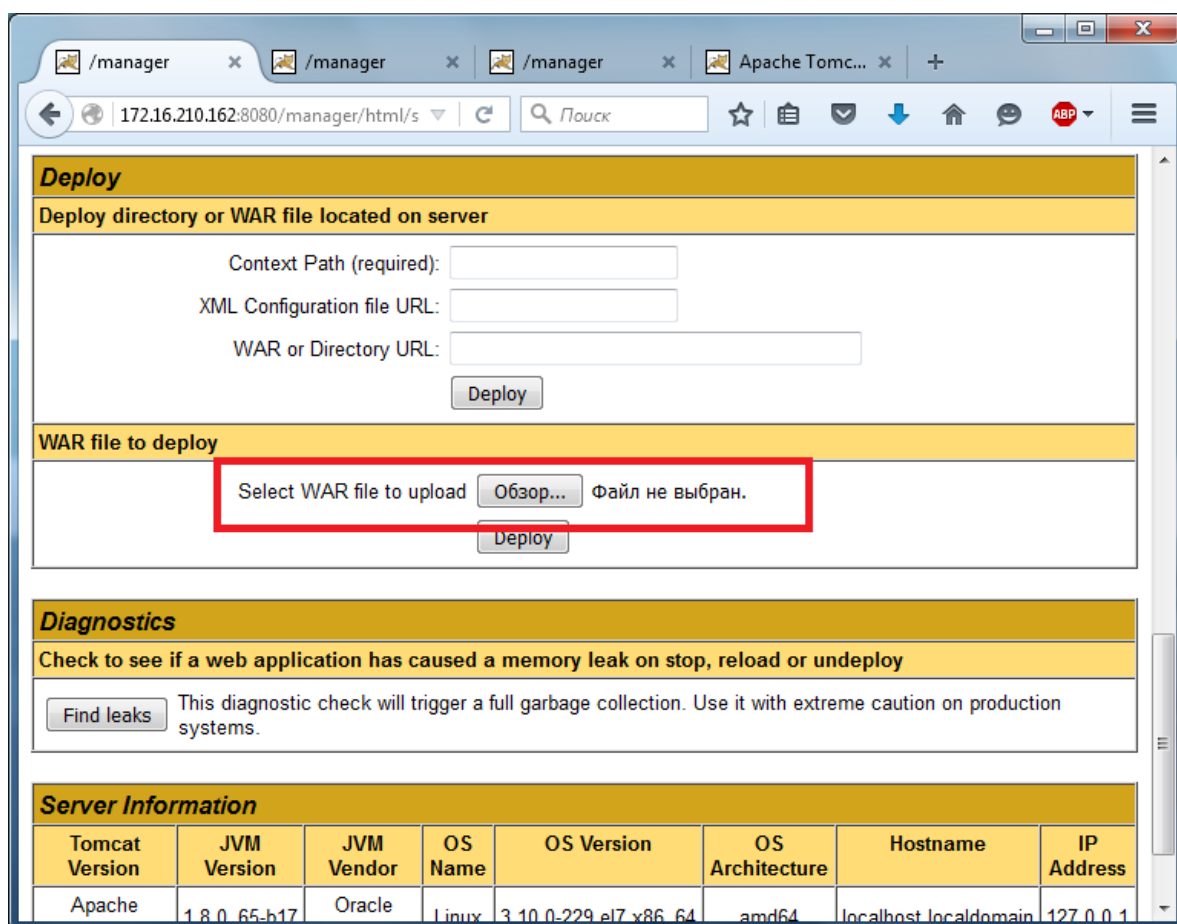


Рисунок 7.1 – Раздел «Deploy» менеджера приложений

После этого Tomcat самостоятельно скопирует war-файл к себе в каталог `\webapps` и развернет его. Web-приложение появится в списке Applications и вы сможете перейти к нему, кликнув по ссылке с соответствующим названием.

8 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ СИСТЕМЫ

Тестирование программного обеспечения – это один из самых важных этапов в процессе создания продукта. Не уделив ему должного внимания, нельзя добиться высокого качества на выходе.

В настоящее время нельзя выделить какой-то конкретный вид тестирования, применение которого гарантирует высокое качество программного продукта. Чтобы добиться хорошего результата, необходимо использовать различные типы тестов [7]. При этом применять стоит как ручные, так и автоматические тесты, комбинирование которых позволит выявить как можно больше ошибок и недочетов. Однако даже использование различных методов тестирования еще не гарантирует того, что в продукте абсолютно не будет ошибок.

Независимо от того, какие методы используются для проверки качества программы и ее соответствия первоначально установленным требованиям, тестировщик обязательно должен изучить и проанализировать следующие факторы: функциональность, эффективность, надежность, мобильность, практичность и сопровождаемость разрабатываемого программного обеспечения или приложения.

Чем позже тестировщики приступят к работам по проекту, тем менее качественным будет продукт. Соответственно, привлекать их к работе стоит уже на стадии обсуждения требований заказчика. Это поможет предотвратить некорректный выбор стратегий разработки продукта и значительно снизить количество возможных ошибок. Если не устранять ошибки на ранних этапах, а делать это после проведения основных работ по проекту, то стоимость всей разработки может возрасти в несколько раз, что абсолютно недопустимо.

На стадии разработки приложение тестировалось несколько раз с помощью ручного и автоматизированного тестирования, используя фреймворк TestNG. Проблемы и ошибки устранялись по мере их выявления. На рисунке 8.1 представлен результат тестирования системы с помощью TestNG. Из 34 тестов все 34 пройдены.

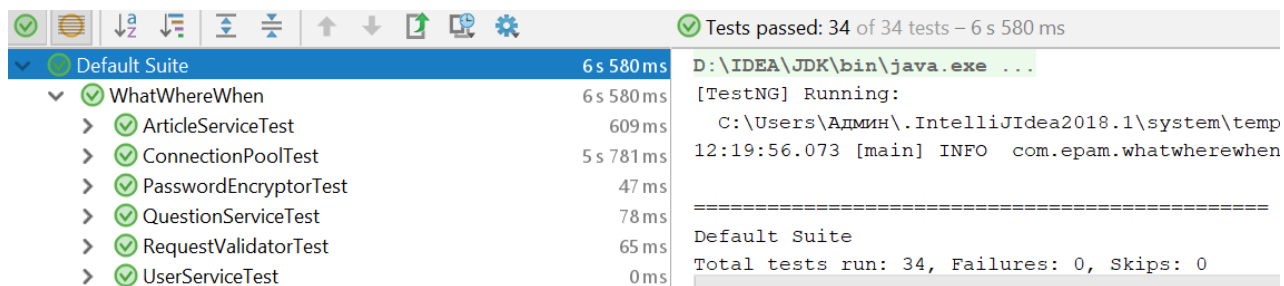


Рисунок 8.1 – Результат тестирования системы

На финальном этапе тестирования с помощью ручного тестирования было подтверждено, что система отвечает предъявленным к ней требованиям, выполняет заложенный в неё функционал.

ЗАКЛЮЧЕНИЕ

Во времена советского телевидения «Что? Где? Когда?» стала первой передачей, приобщавшей людей к интеллектуальным играм. Она задала ритм, а также форматы проведения других интеллектуальных передач.

После оглушительного успеха передачи «Что? Где? Когда» начали организовываться команды, а затем и клубы для игры. Со временем популярность этой игры только нарастала, позже в неё начали играть по всему миру, объединяя между собой людей из разных концов света.

В наши дни ЧГК стала заново возрождать старые традиции, привлекать новых знатоков для игр. Игра помогает не только увеличить объём своих знаний, но и работать в команде, понимать каждого человека, трудиться на получение требуемого результата. В ЧГК важно уметь рассуждать, уметь использовать классические и неклассические методы логики, находить нужные суждения и ответы среди вороха информации, которая получается в период обсуждения вопроса.

По завершению данного курсового проекта был создан портал для любителей «Что? Где? Когда?». Проект создан специально для любознательных людей, которым интересно отвечать на любые, даже самые редкие и специфические вопросы.

Разработка портала любителей «Что? Где? Когда?» позволит добиться следующих результатов:

- повышение интеллектуального и культурного уровня пользователей интернета;
- популяризация форм интеллектуального досуга;
- популяризация и распространение самой игры;
- пропаганда научных знаний и развитие интереса к научной деятельности.

Благодаря хорошей проектировке данной программы в дальнейшем она может редактироваться и совершенствоваться, а главное может быть легко расширяема.

В дальнейшем можно расширить функционал портала с помощью добавления форума игроков, на котором пользователи смогут обмениваться впечатлениями об игре и вопросах. Имеет смысл предоставить возможность пользователям публиковать интересные статьи. Также есть необходимость в совершенствовании системы начисления баллов игрокам и системы проведения соревнований.

На данный момент существует не очень большое количество подобных порталов по теме «Что? Где? Когда». В основном существуют либо игры, либо информационные порталы по заданной теме, поэтому объединение различных функций в одной системе будет удобно и востребовано среди пользователей.

В ходе выполнения данной курсовой работы все поставленные цели были достигнуты. Был изучен процесс проведения спортивной версии игры «Что? Где? Когда?». Также были повышены навыки разработки и проектирования программного обеспечения, уровень владения языком программирования JAVA и технологией Java Servlets. Были приобретены общие теоретические знания о разработке веб-приложений [4].

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- [1] Что? Где? Когда? — Википедия [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://ru.wikipedia.org>
- [2] Кодекс спортивного ЧГК (2018) [Электронный ресурс]. — Электронные данные. — Режим доступа: <http://mak-chgk.ru/rules/codex/>
- [3] Моделирование на UML. Ф.Новиков, Д.Иванов. — Электронные данные. — Режим доступа: http://book.uml3.ru/sec_1_1
- [4] Изучаем Java EE. Современное программирование для больших предприятий. Дашнер С. Издательство: Питер, 2018. — 384 с.
- [5] Java. Методы программирования. И.Н. Блинов, В.С. Романчик Минск: издательство «Четыре четверти», 2013. — 896 с.
- [6] Описание стандарта IDEF0 [Электронный ресурс]. — Электронные данные. — Режим доступа: <http://www.logists.by/library/view/opisanie-standarta-idefo>
- [7] Тестирование. Фундаментальная теория [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://habr.com/ru/post/279535/xhtmll>
- [8] Нормализация отношений. Шесть нормальных форм БД [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://habr.com/post/254773/>
- [9] Apache Tomcat [Электронный ресурс]. — Электронные данные. — Режим доступа: https://wiki.etersoft.ru/Apache_Tomcat
- [10] Обзор паттернов проектирования [Электронный ресурс]. — Электронные данные. — Режим доступа: <http://citforum.ru/SE/project/pattern/>

ПРИЛОЖЕНИЕ А
(обязательное)
Листинг основных элементов программы

ПРИЛОЖЕНИЕ Б **(обязательное)** **Функциональная модель системы**

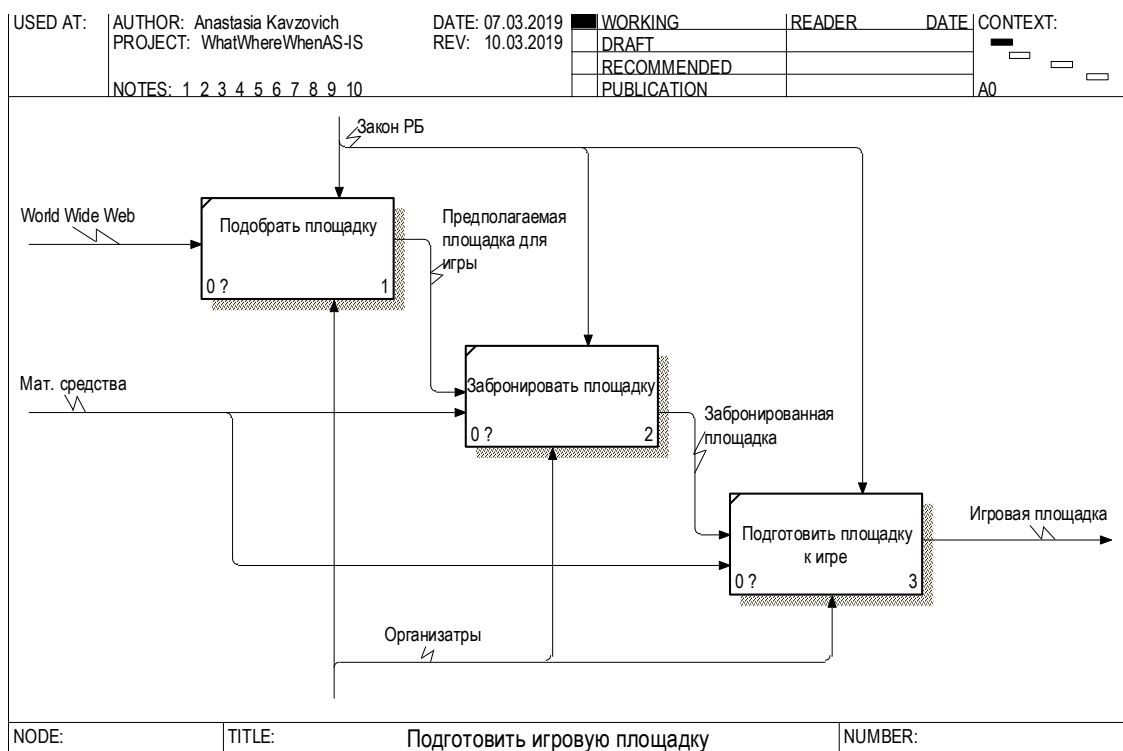


Рисунок Б.1 – Процесс подготовки игровой площадки AS-IS

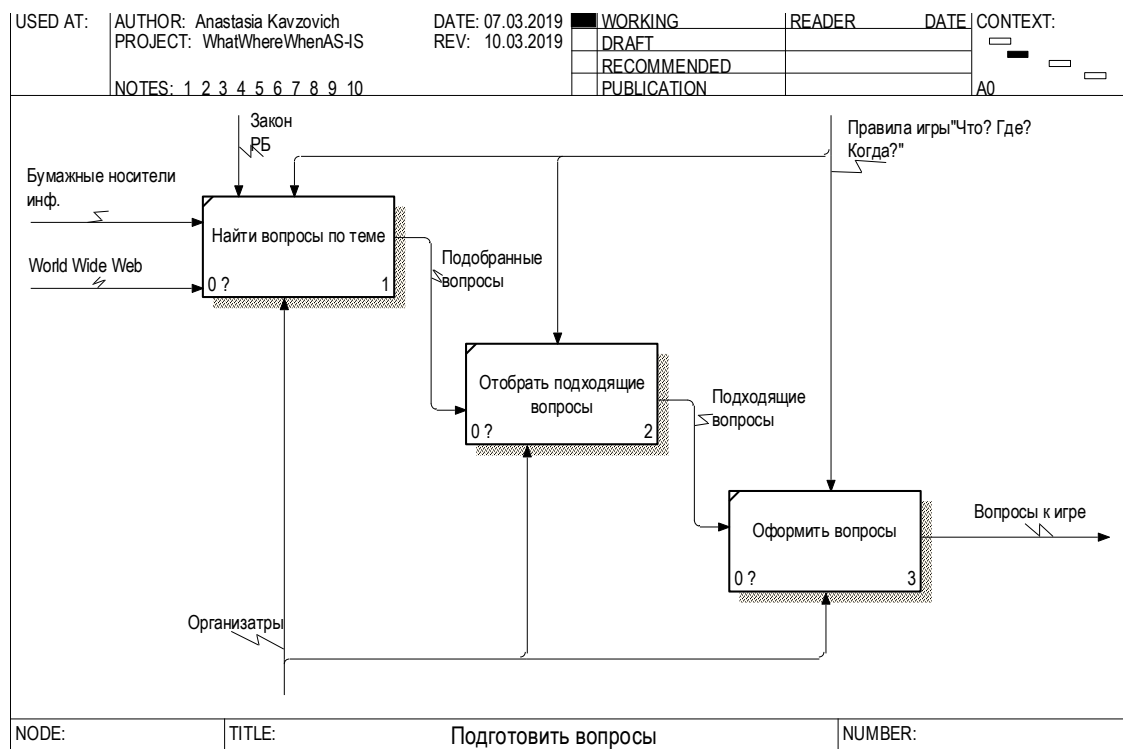


Рисунок Б.2 – Процесс подготовки вопросов AS-IS

Продолжение приложения Б

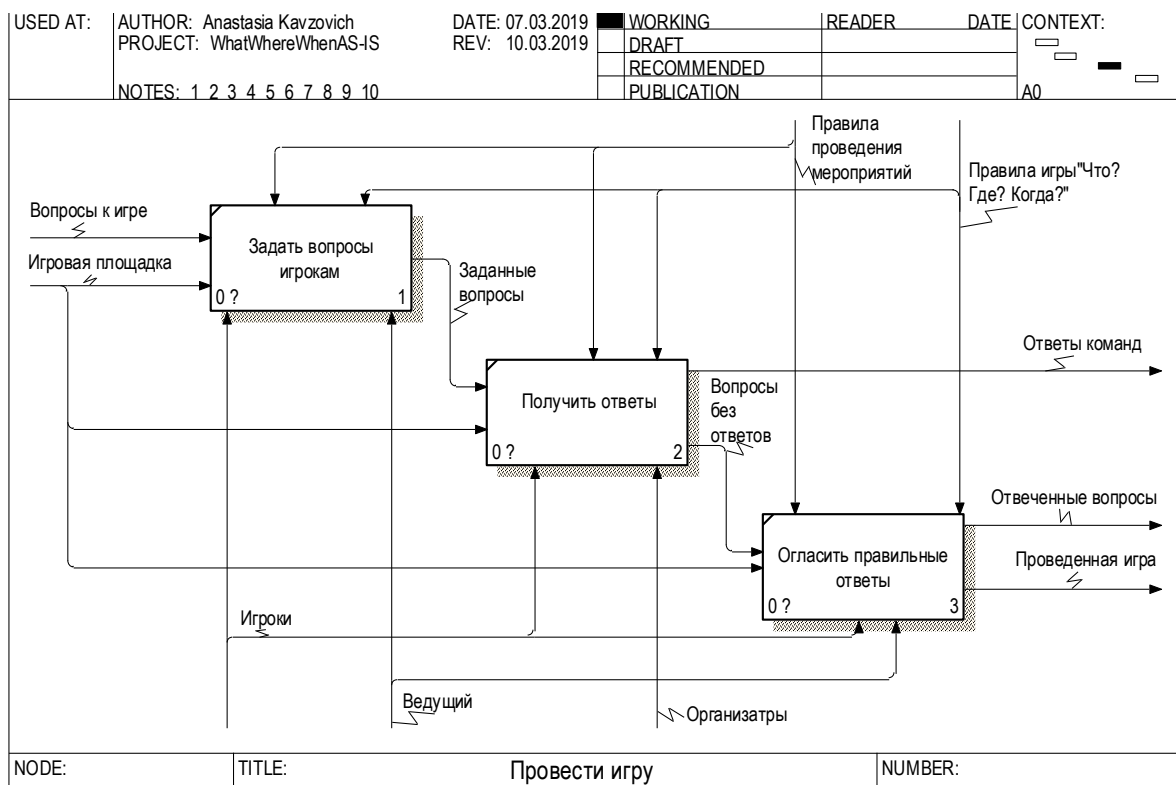


Рисунок Б.3 – Процесс проведения игры AS-IS

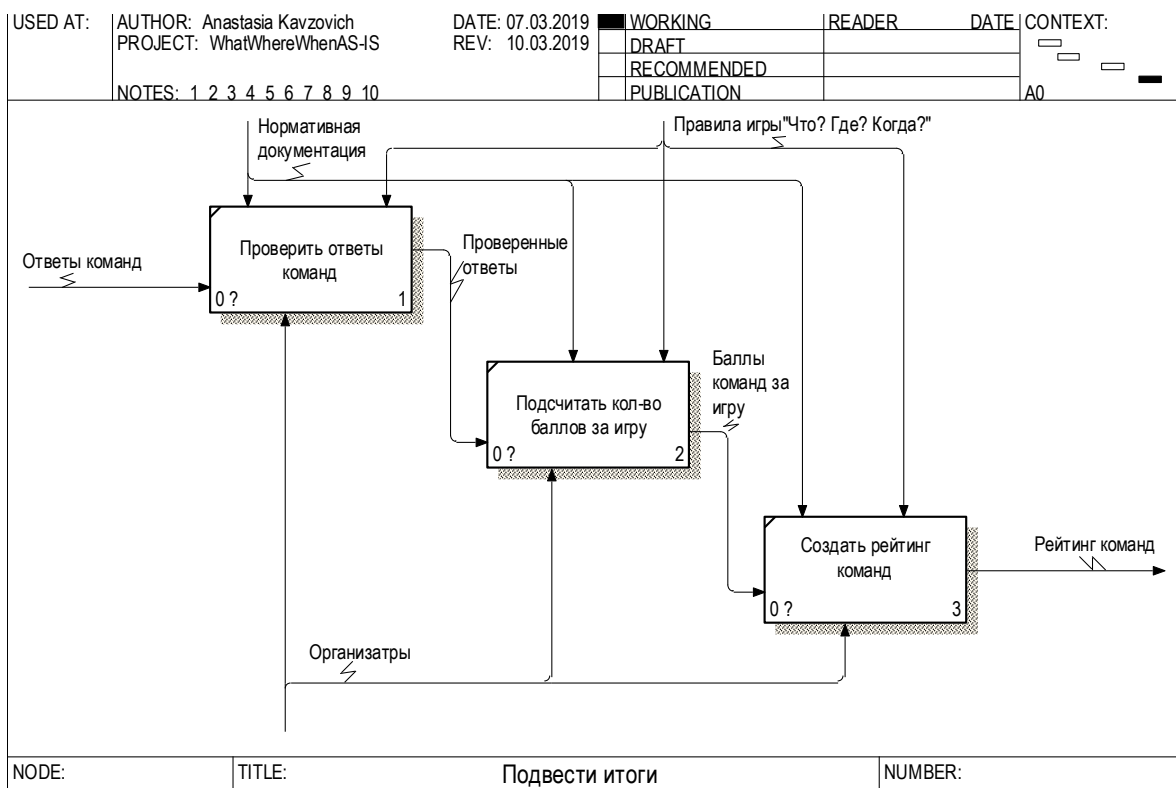


Рисунок Б.4 – Процесс подведения итогов AS-IS

Продолжение приложения Б

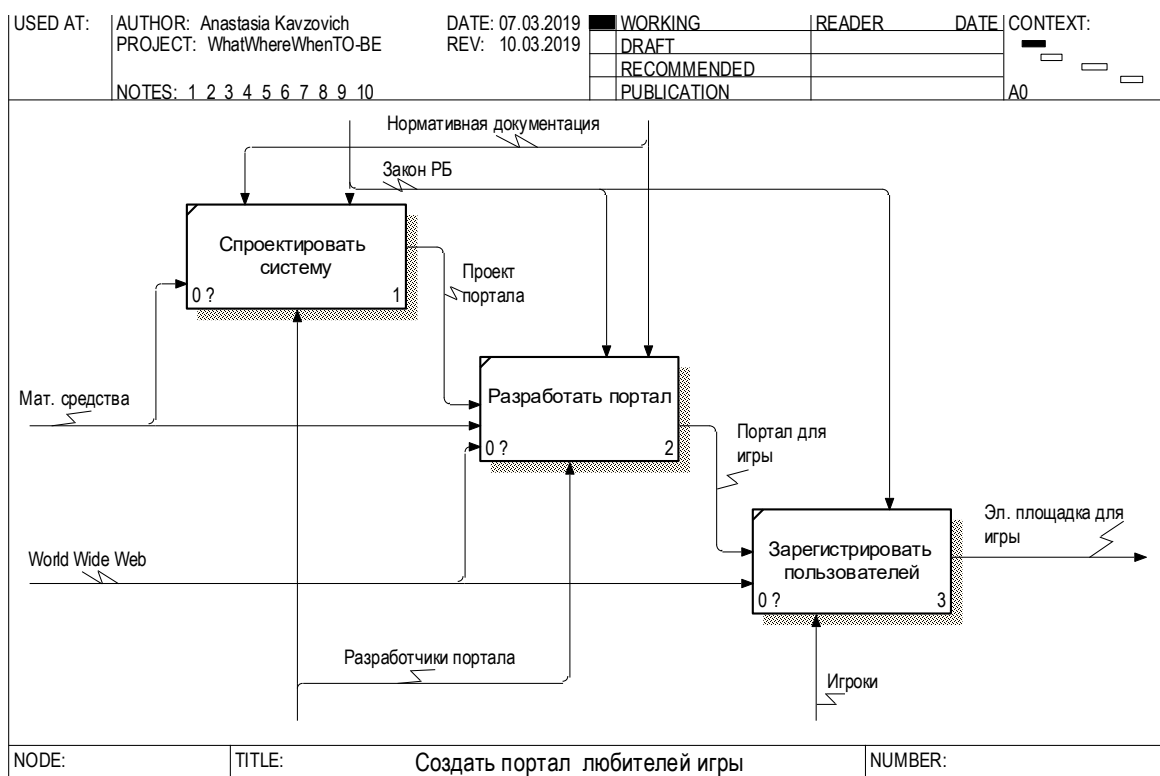


Рисунок Б.5 – Процесс портала любителей игры TO-BE

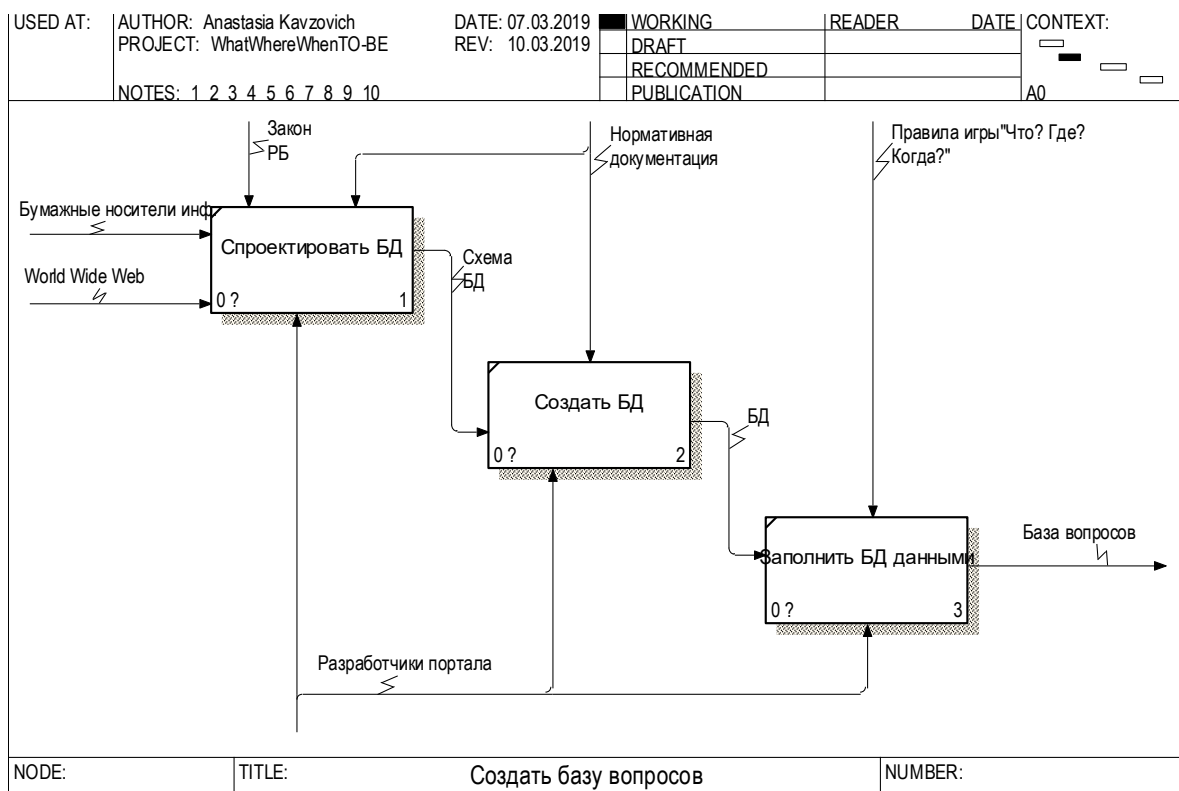


Рисунок Б.6 – Процесс создания базы вопросов TO-BE

Продолжение приложения Б

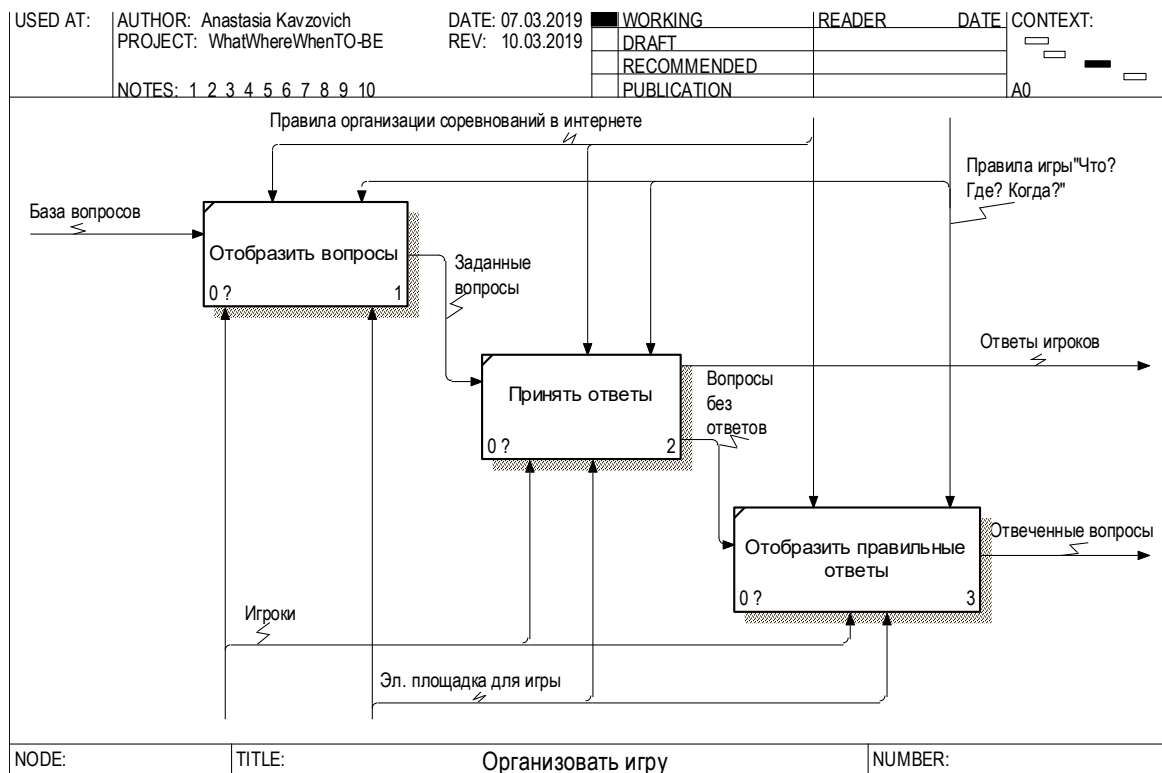


Рисунок Б.7 – Процесс организации игры ТО-ВЕ

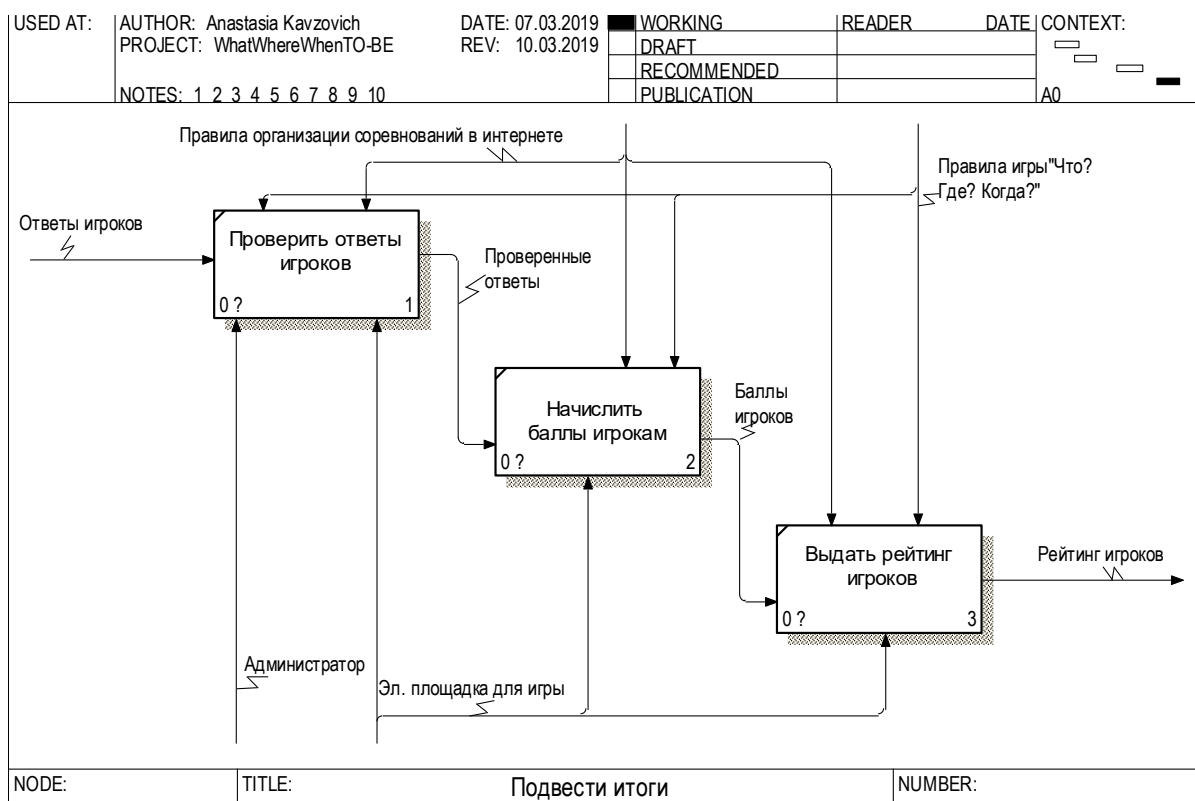


Рисунок Б.8 – Процесс подведения итогов ТО-ВЕ

ПРИЛОЖЕНИЕ В

(обязательное)

Листинг скрипта генерации базы данных

```
CREATE SCHEMA IF NOT EXISTS `whatwherewhen` DEFAULT
CHARACTER SET utf8 ;
USE `whatwherewhen` ;

-----
-- Table `whatwherewhen`.`User`
-----

CREATE TABLE IF NOT EXISTS `whatwherewhen`.`User` (
  `UserId` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `Login` VARCHAR(15) NOT NULL,
  `Password` VARCHAR(15) NOT NULL,
  `Rating` BIGINT NULL DEFAULT '0',
  `IsAdmin` BIT(1) NULL DEFAULT b'0',
  `IsActive` BIT(1) NULL DEFAULT b'1',
  `Photo` BLOB(2048) NULL,
  PRIMARY KEY (`UserId`),
  UNIQUE INDEX `UserId_UNIQUE` (`UserId`),
  UNIQUE INDEX `Login_UNIQUE` (`Login`))
ENGINE = InnoDB
AUTO_INCREMENT = 1
DEFAULT CHARACTER SET = utf8;

-----
-- Table `whatwherewhen`.`Question`
-----

CREATE TABLE IF NOT EXISTS `whatwherewhen`.`Question` (
  `QuestionId` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `AuthorId` BIGINT UNSIGNED NOT NULL,
  `Type` VARCHAR(30) NOT NULL,
  `Body` TEXT(1000) NOT NULL,
  `Answer` TEXT(500) NOT NULL,
  `Date` BIGINT NOT NULL,
  `Source` TEXT(150) NOT NULL,
  `IsActive` BIT(1) NULL DEFAULT b'0',
  `Photo` VARCHAR(45) NULL,
  PRIMARY KEY (`QuestionId`),
  UNIQUE INDEX `QuestionId_UNIQUE` (`QuestionId`),
  INDEX `fk_question_user1_idx` (`AuthorId`),
  CONSTRAINT `fk_question_user1`
```


Продолжение приложения В

```
FOREIGN KEY (`AuthorId`)
REFERENCES `whatwherewhen`.`User` (`UserId`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
-----
-- Table `whatwherewhen`.`UserData`
-----
CREATE TABLE IF NOT EXISTS `whatwherewhen`.`UserData` (
  `UserId` BIGINT UNSIGNED NOT NULL,
  `Email` VARCHAR(45) NOT NULL,
  `FirstName` VARCHAR(30) NOT NULL,
  `LastName` VARCHAR(30) NOT NULL,
  `Birthdate` BIGINT NOT NULL,
  `City` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`UserId`, `Email`),
  UNIQUE INDEX `Email_UNIQUE` (`Email`),
  INDEX `fk_userdata_User_idx` (`UserId`),
  UNIQUE INDEX `UserId_UNIQUE` (`UserId`),
  CONSTRAINT `fk_userdata_user1`
    FOREIGN KEY (`UserId`)
      REFERENCES `whatwherewhen`.`User` (`UserId`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
-----
-- Table `whatwherewhen`.`QuestionStatistics`
-----
CREATE TABLE IF NOT EXISTS `whatwherewhen`.`QuestionStatistic` (
  `QuestionId` BIGINT UNSIGNED NOT NULL,
  `Answered` BIGINT UNSIGNED NOT NULL,
  `NotAnswered` BIGINT UNSIGNED NOT NULL,
  `Liked` BIGINT UNSIGNED NOT NULL,
  `Disliked` BIGINT UNSIGNED NOT NULL,
  PRIMARY KEY (`QuestionId`),
  UNIQUE INDEX `QuestionId_UNIQUE` (`QuestionId`),
  CONSTRAINT `fk_QuestionStatistics_question1`
    FOREIGN KEY (`QuestionId`)
      REFERENCES `whatwherewhen`.`Question` (`QuestionId`)
      ON DELETE NO ACTION
```

Продолжение приложения В

```
ON UPDATE NO ACTION)
ENGINE = InnoDB;
-----
-- Table `whatwherewhen`.`PlayedQuestion`
-----
CREATE TABLE IF NOT EXISTS `whatwherewhen`.`PlayedQuestion` (
  `QuestionId` BIGINT UNSIGNED NOT NULL,
  `UserId` BIGINT UNSIGNED NOT NULL,
  `Answered` BIT NOT NULL,
  `IsOnAppeal` BIT(1) NULL DEFAULT b'0',
  PRIMARY KEY (`QuestionId`, `UserId`),
  INDEX `fk_PlayedQuestion_question1_idx` (`QuestionId`),
  CONSTRAINT `fk_PlayedQuestion_user1`
    FOREIGN KEY (`UserId`)
      REFERENCES `whatwherewhen`.`User` (`UserId`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_PlayedQuestion_question1`
    FOREIGN KEY (`QuestionId`)
      REFERENCES `whatwherewhen`.`Question` (`QuestionId`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
-----
-- Table `whatwherewhen`.`Article`
-----
CREATE TABLE IF NOT EXISTS `whatwherewhen`.`Article` (
  `ArticleId` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `AuthorId` BIGINT UNSIGNED NOT NULL,
  `Theme` VARCHAR(30) NOT NULL,
  `Header` VARCHAR(30) NOT NULL,
  `Body` TEXT(9000) NOT NULL,
  `Date` BIGINT NOT NULL,
  PRIMARY KEY (`ArticleId`),
  INDEX `fk_Article_user1_idx` (`AuthorId`),
  UNIQUE INDEX `ArticleId_UNIQUE` (`ArticleId`),
  CONSTRAINT `fk_Article_user1`
    FOREIGN KEY (`AuthorId`)
      REFERENCES `whatwherewhen`.`User` (`UserId`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

ПРИЛОЖЕНИЕ Г

(обязательное)

Диаграмма классов

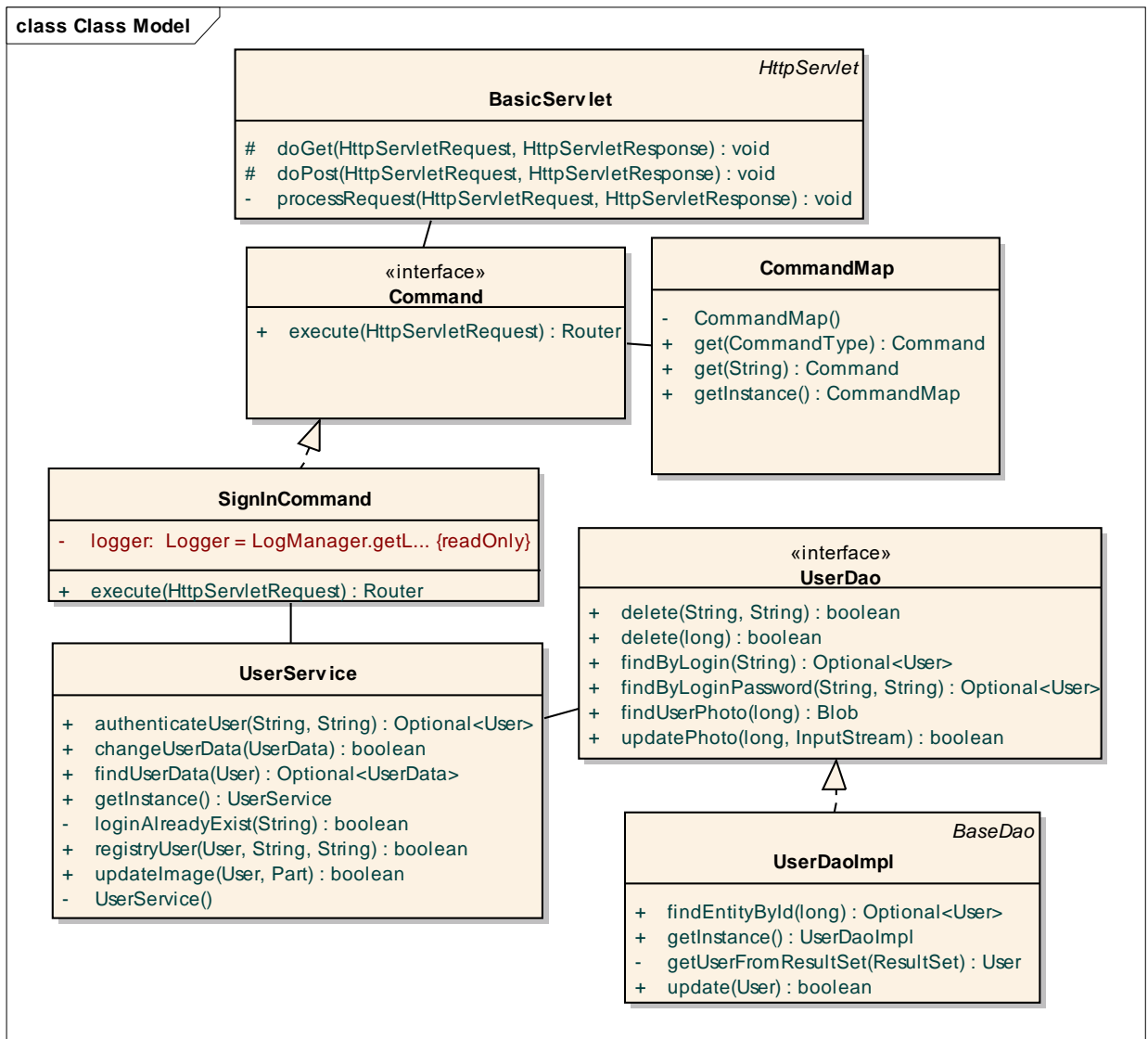


Рисунок Г.1 – Диаграмма классов