МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»  
(Самарский университет)

Институт информатики и кибернетики

Кафедра программных систем

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**«РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ**

**УПРАВЛЕНИЯ ИТ-ПРОЕКТАМИ»**

по направлению подготовки 02.03.02

Фундаментальная информатика и информационные технологии

(уровень бакалавриата)

направленность (профиль) «Информационные технологии»

Обучающийся Г.С. Вазюкова

(подпись, дата)

Руководитель ВКР

к.т.н., доцент Е.В. Сопченко

(подпись, дата)

Нормоконтроллер Е.В. Сопченко

(подпись, дата)

Самара 2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Институт информатики и кибернетики

Кафедра программных систем

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.В. Востокин

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ   
(бакалавр)

обучающейся Вазюковой Галине Сергеевне

группа 6413-020302D

Тема работы: Разработка автоматизированной системы управления

ИТ-проектами

Исходные данные см. в приложении к заданию

Структурные части работы (перечень вопросов, подлежащих разработке):

1. Провести анализ предметной области: управление проектами, методологии Agile, процесс командной разработки
2. Сделать обзор систем-аналогов в области управления ИТ-проектами, реализующих методологии Agile
3. Разработать структурную схему системы
4. Разработать информационно-логический проект системы по методологии UML
5. Разработать интерфейс пользователя
6. Разработать и реализовать программное и информационное обеспечение, провести   
   тестирование и отладку
7. Провести ресурсные расчеты

|  |  |
| --- | --- |
| Научныйруководитель  доцент, кафедра программных систем | Задание принял к исполнению |
| (E.В. Сопченко)  « 26 » октября 2021 г. | (Г.С. Вазюкова)  « 26 » октября 2021 г. |

ПРИЛОЖЕНИЕ

к заданию на выпускную квалификационную работу бакалавра

обучающейся Г.С. Вазюковой группы № 6413-020302D

Тема: «Разработка автоматизированной системы управления ИТ-проектами»

Исходные данные к работе:

1. Характеристики объекта автоматизации:
2. объект автоматизации – процесс управления ИТ-проектами;
3. виды автоматизируемой деятельности:

* процесс создания проекта;
* процесс изменения конфигурации проекта;
* процесс выдачи ролей участников проекта;
* процесс создания задач;
* процесс распределения задач;
* процесс визуализации доски размещения задач;

1. максимальное количество проектов – 100;
2. максимальная длина фамилии врача – 30 символов;
3. максимальная длина имени врача – 30 символов;
4. максимальная длина отчества врача – 30 символов;
5. максимальная длина логина врача – 15 символов;
6. минимальная длина пароля врача – 4 символа;
7. минимальное количество пациентов – 100;
8. максимальная длина фамилии пациента – 30 символов;
9. максимальная длина имени пациента – 30 символов;
10. максимальная длина отчества пациента – 30 символов;
11. количество групп пациентов – 6;
12. количество параметров осмотра – 230;
13. количество статистических графиков – 6;
14. количество географических справочников – 5;
15. максимальная длина названия элемента справочника – 30 символов.
16. Требования к информационному обеспечению:
17. Информационное обеспечение разрабатывается на основании следующих литературных источников:

* Покровская О.М. Совершенствование комплекса гигиенических мероприятий у пациентов с ортопедическими конструкциями на имплантатах. : дис. ... канд. мед. наук: 14.00.21: защищена 02.09.2008/ Покровская Ольга Михайловна. М., 2008. 115 с.: ил. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. СПб.: Питер, 2018. 480 с.
* Справочник по формулам зубов [Электронный ресурс]. URL: http://www.dental-revue.ru/index.php?page=15&artId=7 (дата обращения: 30.09.2018).
* Современные системы оценки и регистрации кариеса зубов / Пастбин М.Ю., Горбатова М.А. и [др.]. Архангельск: Северный государственный медицинский университет, 2013. 7 с.
* Методическое пособие по теме «Индексная оценка кариеса зубов и заболеваний пародонта» / Молоков В.Д., Доржиева З.В. и [др.]. Иркутск: Иркутский государственный медицинский университет, 2008. 23 с.

1. Структура базы данных разрабатывается на основании следующих сведений:

* о врачах (ФИО, логин, пароль);
* о пациентах (ФИО, пол, дата рождения, группа, район проживания, адрес);
* об осмотрах (имя параметра, значение параметра);
* о справочниках (название элемента справочника).

1. Обеспечить контроль целостности базы данных.
2. Требования к техническому обеспечению:
   1. Требования к техническому обеспечению серверной части:
3. тип ЭВМ – IBM PC совместимый;
4. объем ОЗУ – не менее 2 Гб;
5. объем свободного пространства на внешнем диске – не менее   
   50 Гб;
6. наличие подключения к сети Интернет;
7. манипулятор – мышь;
8. технические характеристики определяются в процессе выполнения работы.
   1. Требования к техническому обеспечению клиентской части:
9. тип ЭВМ – IBM PC совместимый;
10. монитор с разрешающей способностью не ниже 800 х 600;
11. манипулятор – мышь;
12. технические характеристики определяются в процессе выполнения работы.
13. Требования к программному обеспечению:
    1. Требования к программному обеспечению серверной части:
14. тип операционной системы – Windows 7 и выше;
15. СУБД – PostgrateSQL 10;
    1. Требования к программному обеспечению клиентской части:
16. тип операционной системы – Windows 7 и выше;
17. браузер – Google Chrome 86.0.4240.183 (64-битный) и выше, Firefox 83.0 (64-битный) и выше.
    1. Требования к программному обеспечению рабочего места разработчика:
18. тип операционной системы – Windows 7 и выше;
19. язык программирования – С#;
20. среда программирования – Visual Studio 2019;
21. СУБД – PostgreSQL 10;
22. среда проектирования – StarUML 4.0.0.
23. Общие требования к проектируемой системе.
    1. Функции, реализуемые системой:
24. Общесистемные функции:

* ведение базы данных;
* вычисление стоматологических индексов пациентов;
* выдача справочной информации.

1. Функции врача:

* авторизация врача:

1. ввод логина;
2. ввод пароля;

* регистрация врача:

1. ввод ФИО;
2. ввод логина;
3. ввод пароля;

* ввод данных о пациенте;
* редактирование списка пациентов;
* ввод данных осмотра пациента;
* редактирование списка осмотров у выбранного пациента;
* вывод результатов осмотров в виде таблицы Excel для всех пациентов;
* вывод результатов осмотра в виде карты осмотра в документ Word для отдельного пациента;
* просмотр статистических графиков на основе результатов осмотров.
  1. Технические требования к системе:

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

Научный руководитель,

к.т.н., доцент Е.В. Сопченко

подпись, дата

РЕФЕРАТ

Пояснительная записка: 101 с, 55 рисунков, 10 таблиц, 31 источник, 5 приложений.

Графическая часть: 34 слайда презентации PowerPoint.

КЛЮЧЕВЫЕ СЛОВА НЕ МЕНЕЕ 6-7 СЛОВ ИЛИ СЛОВОСОЧЕТАНИЙ

Объектом исследования является ….

Цель работы – разработать автоматизированную систему ….

В процессе работы были разработаны алгоритмы и соответствующая программа, позволяющая пользователю …. Система позволяет ….

Система разработана на языке ХХХ с использованием фреймворка ХХ, библиотек ХХХ, ХХХ, ХХХ и функционирует под управлением операционных систем Windows 7/8/10. Доступ к данным осуществляется с помощью СУБД SQLite 3.24.

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

|  |  |  |
| --- | --- | --- |
| API | − | Application programming interface (программный интерфейс приложения) |
| REST | − | Representational State Transfer (передача репрезентативного состояния) |
| БД | − | база данных; |
| СУБД | − | система управления базами данных; |

Содержание

[ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 9](#_Toc99902544)

[Введение 12](#_Toc99902545)

[1 Описание и анализ предметной области 13](#_Toc99902546)

[1.1 Работа в команде 13](#_Toc99902547)

[1.2 Методологии управления проектами 13](#_Toc99902548)

[1.2.1 Каскадная методология управления проектами (Waterfall) 13](#_Toc99902549)

[1.2.2 Методология управления проектами Agile 14](#_Toc99902550)

[1.2.2.1. Scrum 14](#_Toc99902551)

[1.2.2.2. Kanban 14](#_Toc99902552)

[1.3 Обзор существующих систем-аналогов 16](#_Toc99902553)

[1.3.3 JIRA 16](#_Toc99902554)

[1.3.4 Trello 17](#_Toc99902555)

[1.4 Диаграмма объектов предметной области 20](#_Toc99902556)

[1.5 Постановка задачи 20](#_Toc99902557)

[1.5.1 Режим системного администратора 21](#_Toc99902558)

[1.5.2 Режим менеджера проекта 22](#_Toc99902559)

[1.5.3 Режим участника проекта 22](#_Toc99902560)

[2 Проектирование системы 23](#_Toc99902561)

[2.1 Выбор и обоснование архитектуры системы 23](#_Toc99902562)

[2.2 Структурная схема системы 23](#_Toc99902563)

[2.3 Разработка прототипов экранных форм приложения 23](#_Toc99902564)

[2.4 Разработка информационно-логического проекта системы 24](#_Toc99902565)

[2.4.1 Язык UML 24](#_Toc99902566)

[2.4.2 Диаграмма вариантов использования 24](#_Toc99902567)

[2.4.3 Сценарии 26](#_Toc99902568)

[2.4.4 Диаграмма классов 26](#_Toc99902569)

[2.4.5 Диаграмма деятельности 27](#_Toc99902570)

[2.4.6 Логическая модель данных системы 27](#_Toc99902571)

[2.5 Выбор и обоснование комплекса программных средств 29](#_Toc99902572)

[2.5.1 Выбор языка программирования 29](#_Toc99902573)

[2.5.2 Выбор операционной системы 29](#_Toc99902574)

[2.5.3 Выбор среды программирования 30](#_Toc99902575)

[2.5.4 Выбор СУБД 30](#_Toc99902576)

[Заключение 32](#_Toc99902577)

[Список использованных источников 33](#_Toc99902578)

Введение

В наше время разработка и ведение проектов часто осуществляется в команде. Для грамотной реализации командной работы проектные менеджеры часто прибегают к использованию методологии организации работы в команде. Методологии управления проектами представляют собой разные подходы к организации проекта. Следуя четким правилам по организации разработки, команда может:

* четко поставить цель и правильно ее достичь;
* корректно распределить задачи между участниками проекта;
* выделять более приоритетные задачи для выполнения;
* следить за продвижением проекта в целом.

Наиболее удобным способом использования методологий являются специализированные системы, которые помогают избавиться от ведения дел на бумаге. При помощи этих приложений работа в команде упрощается не только для руководителя, но и для участников проекта, т.к. каждому из них четко понятны все задачи и требования, которые необходимо выполнить.

Также в настоящее время приобретают большую популярность мобильные приложения по ряду причин:

* Отсутствие необходимости иметь персональный компьютер для доступа к приложению;
* Удобство использования – получить необходимый функционал можно в любое время;
* Рост популярности мобильных устройств.

В ходе выполнения выпускной квалификационной работы необходимо разработать мобильное приложение для организации и ведения работы в команде.

1. Описание и анализ предметной области
   1. Работа в команде

Разработка проектов зачастую ведется в команде. На это есть несколько причин:

* разработка проекта осуществляется гораздо быстрее;
* командная разработка представляет собой одну из форм делегирования полномочий.
* командная работа наделяет людей способностью к сотрудничеству.

Чтобы работа команды была успешной, необходимо соблюдать следующие условия:

* грамотная постановка целей;
* наличие четких и ясных задач;
* правильный подбор командного состава;
* наличие детализированной системы работы.

Системы планирования, постановки и контроля задач — основной инструмент работы над проектами, который поможет в выполнении этих условий.

* 1. Методологии управления проектами
     1. Каскадная методология управления проектами (Waterfall)

Самый распространенный способ планирования проектов – составление последовательности задач, приводящей к конечному результату, и поочередное выполнение этих задач. Так выглядит каскадная методология — традиционный и наиболее простой для понимания метод управления проектами. Предыдущая задача должна быть выполнена до начала выполнения следующей задачи в составе связной последовательности элементов, в совокупности приводящих к конечному результату.

Достоинства:

Недостатки:

* + 1. Методология управления проектами Agile

Ключевые принципы методологии Agile были разработаны в 2001 году и включают четыре основных ценности:

* люди и взаимодействие важнее процессов и инструментов;
* работающий продукт важнее исчерпывающей документации;
* сотрудничество с клиентом важнее согласования условий контракта;
* готовность к изменениям важнее следования первоначальному плану.
  + - 1. Scrum

Scrum — наиболее популярный фреймворк методологии Agile за счет относительной простоты реализации. Он также позволяет решить множество проблем, с которыми ранее сталкивались разработчики программного обеспечения: например, запутанность циклов разработки, отсутствие гибкости планов проектов и сдвиги производственных графиков.

В методологии Scrum работу небольшой команды организует скрам-мастер, чья основная задача — устранить все препятствия на пути более эффективного выполнения работы. Рабочий процесс команды делится на короткие двухнедельные циклы, называемые «спринты», при этом участники команды ежедневно обсуждают проделанную работу и помехи, которые необходимо устранить. Эта методология ускоряет разработку и тестирование, особенно в небольших командах.

Достоинства

Недостатки

* + - 1. Kanban

Kanban — еще один фреймворк внедрения Agile-методологий, основанный на уровне загрузки команды. Этот метод был разработан на заводах компании Toyota в 1940-х годах и изначально представлял собой визуальную систему карточек («канбанов»), используемых отделами в качестве сигналов о готовности команды к обработке следующей партии сырья, о способности команды производить больше.

В наше время этот визуальный подход к управлению проектом отлично подходит для работы, требующей постоянной отдачи. Проектные группы создают визуализацию своих задач, зачастую при помощи стикеров и досок и двигают стикеры с задачами от первой до последней заранее установленной стадии, чтобы отслеживать прогресс в ходе выполнения задач и выявлять препятствия.

На рисунке N представлен пример реализации Kanban-доски. Все проекты в ней разделяются по статусу разработки. Также присутствует список потенциальных проектов. Методика планирования подчиняется ряду правил:

1. списки состояний первоначально пусты, заполнен лишь список потенциальных проектов;
2. проекты, которые необходимо взять в разработку перемещаются в первое состояние – начала разработки;
3. в ходе разработки проекта он перемещается в следующий по порядку столбец доски при достижении определенного состояния;
4. при завершении работы проект пропадает с доски и его статус становится «Закрыт».

  
Рисунок N – Пример реализации Kanban-доски

Достоинства

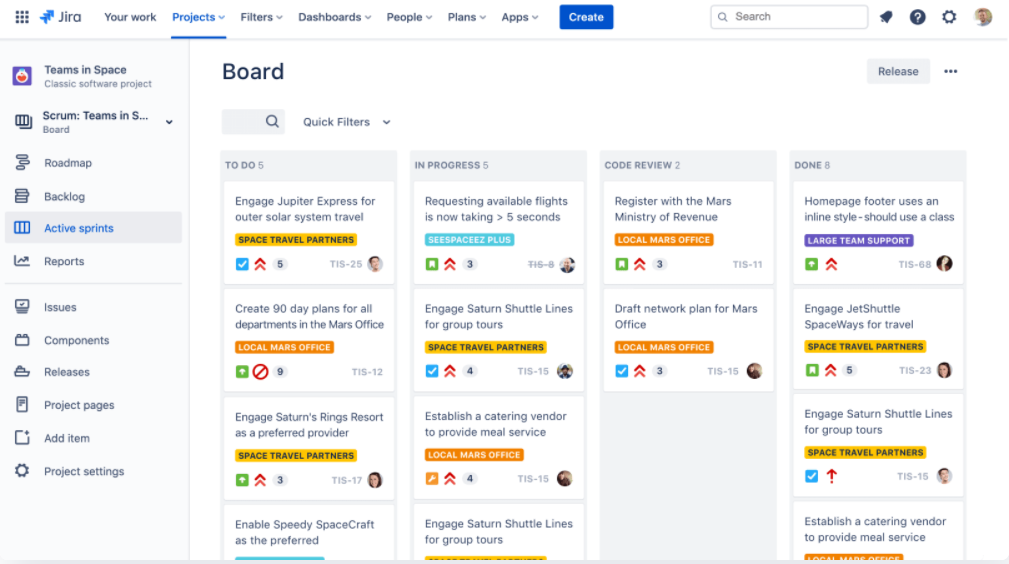
Недостатки

* 1. Обзор существующих систем-аналогов

В настоящее время реализовано множество приложений для организации работы в команде, использующей методологию Kanban. Их анализ поможет подчеркнуть важные детали, на которые нужно обратить внимание при реализации системы.

* + 1. JIRA

Atlassian Jira Software – профессиональный инструмент разработки и управления проектами для agile-команд. Решение позволяет создавать пользовательские истории и задачи, планировать спринты и распределять задания в своей команде разработчиков. Можно использовать стандартный процесс или создавать свой собственный, подходящий именно конкретной команде[].  На рисунке N приведена главная экранная форма программы «JIRA», на которой представлен пример реализации доски по методологии Kanban.

  
Рисунок 2 – Экранная форма программы «JIRA»

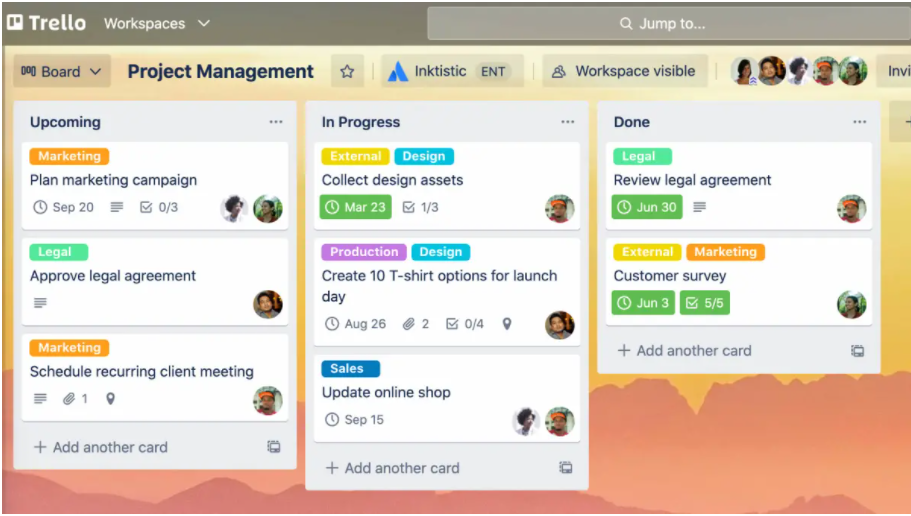
К достоинствам данной системы относятся:

* возможность реализации собственной методологии управления проектом;
* возможность интеграции с другими сервисами и отслеживания работы – системы code-review, документации и т.д.;
* возможность просматривать отчеты по итогам работы команды;
* возможность кастомизации при помощи загрузки сторонних плагинов.

К недостаткам системы относятся:

* сложность в освоении;
* сложность в технической поддержке;
* тяжеловесность;
* доступ к использованию функционала по платной подписке;
* неудобство работы в мобильной версии.
  + 1. Trello

Trello — это одна из самых популярных систем управления проектами в режиме онлайн, которая пользуется особенным спросом среди небольших компаний и стартапов. Она позволяет эффективно организовывать работу по японской методологии Kanban-досок[]. На рисунке N приведена главная экранная форма программы «Trello», на которой приведен пример реализации Kanban-доски.

  
Рисунок N – Экранная форма программы «Trello»

К достоинствам данной системы относятся:

* простой интерфейс;
* почти неограниченный бесплатный доступ;
* возможность интеграции с другими популярными инструментами для онлайн-работы;

К недостаткам системы относятся:

* интерфейс становится неудобным при ведении нескольких проектов сразу;
* нет возможности добавлять описание проектов и модифицировать подзадачи;
* неудобство работы в мобильной версии.

В таблице 1 приведена сравнительная характеристика рассмотренных систем по нескольким параметрам.

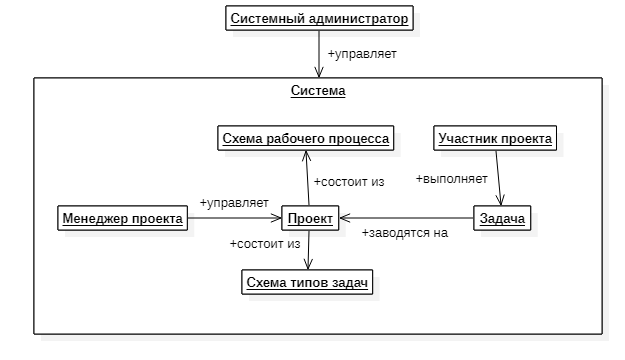
Таблица 1 – Сравнительная характеристика функциональности систем-аналогов

|  |  |  |  |
| --- | --- | --- | --- |
| Название показателя | JIRA | Trello | Разрабатываемая система |
| Удобство использования на мобильном устройстве | Нет | Нет | Да |
| Простота в освоении | Нет | Да | Да |
| Тяжеловесность | Да | Нет | Нет |
| Простой интерфейс | Нет | Да | Да |
| Простота интерфейса при ведении нескольких проектов | Да | Нет | Да |
| Возможность модификации задач | Да | Нет | Да |
| Возможность добавления описания проекта | Да | Нет | Да |
| Возможность выдачи различных прав доступа к проекту | Да | Да | Да |

* 2. Диаграмма объектов предметной области

В предметной области системы управления проектами можно выделить такие основные объекты, как проект, задачи, которая создается на проекте, схема рабочего процесса, схема типов задач – из которых создается конфигурация проекта, участник проекта, который выполняет задачи, менеджер, который управляет проектом и системный администратор, который управляет системой.

На рисунке N приведена диаграмма объектов предметной области. Далее описать основные характеристики объектов.

  
Рисунок N – Диаграмма объектов предметной области

* 1. Постановка задачи

В результате выполнения выпускной квалификационной работы необходимо разработать мобильное приложение, с помощью которого можно управлять работой в команде: создавать проекты, создавать и редактировать задачи в них, отслеживать процесс выполнения задач и работы каждого из участников проекта.

В системе должно быть реализовано три роли: системный администратор, проектный менеджер и участник проекта. Для каждой роли будет предусмотрена авторизация с использованием логина (длина от 6 до 12 символов), пароля (длина от 8 до 16 символов) и электронной почты.

Данные о пользователях, проектах, схемах проектов и задачах будет храниться в базе данных.

* + 1. Режим системного администратора

Системный администратор – пользователь, наделенный наибольшим количеством прав доступа к приложению.

* основные задачи системного администратора:
* создание и настройка конфигурации проектов;
* создание схем проектов;
* выдача прав для других пользователей приложения.

Конфигурация проекта настраивается на основе схем, которые создает системный администратор. В системе будет использоваться 2 вида схем для настройки проекта:

* схема рабочего процесса – набор статусов, по которым будет перемещаться задача;
* схема приоритетов – набор приоритетов, по которым задачи будут выстраиваться по иерархии – от наиболее приоритетных к менее приоритетным.

На проекте должны быть настроены две схемы разных типов. Также на проекте не может быть настроено две схемы одного типа.

Также в приложении будет реализована подсистема обращения к системному администратору других пользователей. Она будет реализована в виде отдельного проекта, в который смогут заводить задачи все пользователи в независимости от роли. Проект будет настроен по особенным схемам:

* схема рабочего процесса будет иметь всего два статуса – задача открыта и выполнена;

схема приоритетов включает в себя один вид приоритета.

* + 1. Режим менеджера проекта

Проектный менеджер – пользователь, который будет управлять конкретным проектом. Он будет иметь возможность редактирования проекта, на который он назначен, при этом к другим проектам у него будет доступ только для просмотра.

Основные задачи проектного менеджера:

* редактирование проекта – изменение описания, добавления участников в проект;
* создание новых задач внутри проекта;
* подтверждение завершения работы над задачей;
* выставление приоритетов задач;
* назначение задачи на участника проекта.
  + 1. Режим участника проекта

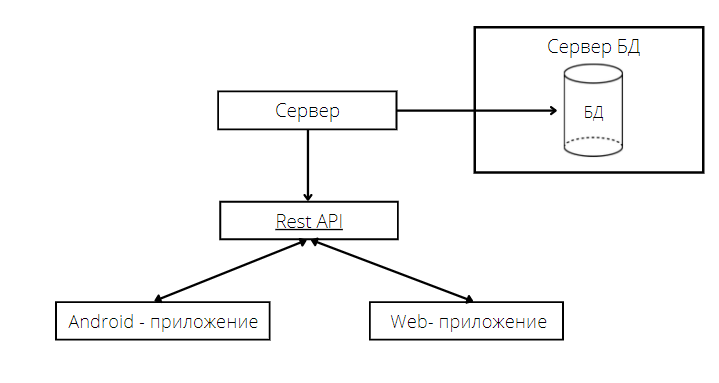
Участник проекта – пользователь, наделенный наименьшим количеством прав для управления системой. Основная задача участника – перемещение задачи из одного состояния в другое.

1. Проектирование системы
   1. Выбор и обоснование архитектуры системы

Архитектура системы — принципиальная организация системы, воплощенная в её элементах, их взаимоотношениях друг с другом и со средой, а также принципы, направляющие её проектирование и эволюцию [].

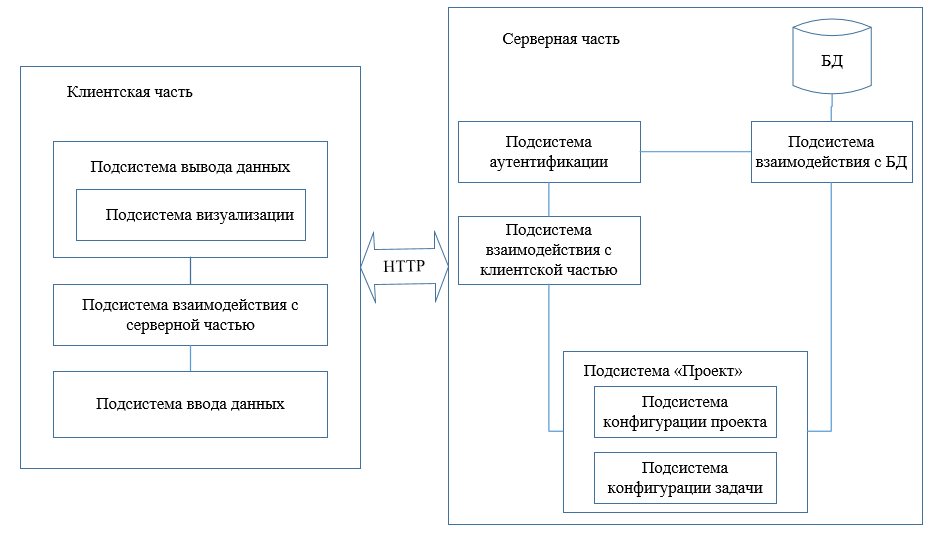
Для реализации приложения была выбрана клиент-серверная трехзвенная архитектура с двумя видами клиентов – мобильное приложение для менеджеров и участников и веб-приложение для системных администраторов. Логика обработки данных будет располагаться на сервере, который будет иметь REST API (интерфейс для взаимодействия клиентов и сервером). Данные в свою очередь будут храниться в базе данных, которая расположена на сервере БД.

На рисунке N приведена структурная схема реализации архитектуры разрабатываемой системы.

  
Рисунок N – Структурная схема реализации архитектуры системы

* 1. Структурная схема системы

На рисунке 5 приведена структурная схема разрабатываемой системы. Система разделена на две части (клиентская и серверная части, которые обмениваются информацией через HTTP-протокол).

  
Рисунок N – Структурная схема системы

В состав клиентской части входят следующие подсистемы:

1. подсистема вывода данных, в состав которой входит подсистема визуализации, которая отвечает за вывод информации на экранную форму приложения;
2. подсистема ввода данных, которая отвечает за ввод информации с форм приложения;
3. подсистема взаимодействия с серверной частью, которая отвечает за взаимодействие с серверной частью.

В состав серверной части приложения входят следующие подсистемы:

1. подсистема аутентификации, которая отвечает за аутентификацию пользователей;
2. подсистема взаимодействия с БД, которая отвечает за взаимодействие системы с БД;
3. подсистема взаимодействия с серверной частью, которая отвечает за взаимодействие серверной части с клиентской;
4. подсистема «Проект», в состав которой входят следующие подсистемы:
   * подсистема конфигурации проекта, которая отвечает за создание и изменение конфигурации проекта;
   * подсистема конфигурации задачи, которая отвечает за создание и изменение задачи.
   1. Разработка прототипов экранных форм приложения

Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова. Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова.

* 1. Разработка информационно-логического проекта системы
     1. Язык UML

Унифицированный язык моделирования (Unified Modeling Language – UML) – это стандартный инструмент для разработки «чертежей» программного обеспечения. Его можно использовать для визуализации, спецификации, конструирования и документирования артефактов программных систем. UML подходит для моделирования любых систем – от информационных систем масштаба предприятия до распределенных Web-приложений и даже встроенных систем реального времени [10].

В данной работе будем использовать UML для специфицирования системы и ее документирования. Язык UML предоставляет стандартный способ написания проектной документации на системы, включая концептуальные аспекты, такие как бизнес-процессы и функции системы, а также конкретные аспекты, такие как выражения языков программирования, схемы баз данных и повторно используемые компоненты ПО [].

* + 1. Диаграмма вариантов использования

Диаграмма вариантов использования представляет собой наиболее общую концептуальную модель сложной системы, которая является исходной для построения всех остальных диаграмм. На ней изображаются отношения между актерами и вариантами использования [].

Актер (actor) – согласованное множество ролей, которые играют внешние сущности по отношению к вариантам использования при взаимодействии с ними.

Вариант использования – внешняя спецификация последовательности действий, которые система или другая сущность могут выполнять в процессе взаимодействия с актерами [].

Цель спецификации варианта использования заключается в том, чтобы зафиксировать некоторый аспект или фрагмент поведения проектируемой системы без указания особенностей реализации данной функциональности. В этом смысле каждый вариант использования соответствует отдельному сервису, который предоставляет моделируемая система по запросу актера, т. е. определяет один из способов применения системы. Сервис, который инициализируется по запросу актера, должен представлять собой законченную последовательность действий. Это означает, что после того как система закончит обработку запроса актера, она должна возвратиться в исходное состояние, в котором снова готова к выполнению следующих запросов [].

На рисунке представлена диаграмма вариантов использования общей части системы для пользователя. Описание диаграммы.

Рисунок ХХХ – Диаграмма вариантов использования системы   
(со стороны пользователя)

На рисунке ХХХ представлена диаграмма вариантов использования ххх.

Рисунок ХХХ – Диаграмма вариантов использования системы   
(со стороны системы)

* + 1. Диаграмма классов

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

На рисунке приведена диаграмма классов системы (этап проектирования). В таблице приведено описание классов.

Таблица 8 – Описание классов системы (этап проектирования)

|  |  |
| --- | --- |
| Название класса | Назначение |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* + 1. Диаграмма деятельности

Диаграмма деятельности – это диаграмма, применяемая в UML для моделирования динамических аспектов систем. По сути, диаграмма деятельности представляет собой блок-схему, которая показывает, как поток управления переходит от одной деятельности к другой.

Моделирование динамических аспектов систем при помощи диаграмм деятельности большей частью подразумевает моделирование последовательных шагов вычислительного процесса. Диаграммы деятельности могут использоваться отдельно для визуализации, специфицирования, конструирования и документирования динамики сообщества объектов либо для моделирования потока управления в операции. Деятельность – это структурированное описание текущего поведения [10].

На рисунке 22 приведена диаграмма деятельности добавления …+описание.

* + 1. Логическая модель данных системы

Логическая информационная модель – модель данных, в которой учитывается способ логического хранения данных в памяти ЭВМ. При построении модели базы данных (БД) используются следующие понятия.

Сущность – объект предметной области, который можно отличить от других понятий по некоторым признакам. Сущность состоит из множества своих экземпляров. Каждая сущность обладает свойствами – атрибутами [].

Атрибут – определенное свойство сущности. Именно набор атрибутов, в общем случае уникальный для каждой сущности, позволяет выделить ее среди других объектов и назвать уникальным именем.

Атрибут или набор атрибутов, используемый для идентификации экземпляра сущности, называется ключом сущности. В случае если для идентификации экземпляра используется один атрибут, ключ называется простым; в противном случае ключ составной. Каждый экземпляр сущности однозначно определяется ключом [].

Логическая модель БД разрабатываемой системы приведена на рисунке ХХХ.

  
Рисунок ХХХ – Логическая модель данных

Описание объектов рассматриваемой предметной области, которые хранятся в базе данных, приведено в таблицах 2-???.

Таблица 2 – Сущность «Пользователь»

|  |  |  |
| --- | --- | --- |
| Идентификатор | Тип данных | Описание |
| Ид пользователя | Целый | Уникальный идентификатор пользователя |
| Имя | Символьный[30] | Имя, используемое при идентификации пользователя и его взаимодействии с системой |
| Пароль | Символьный[10] | Пароль пользователя, преобразованный в закодированную строку |
| Email | Символьный[50] | Электронная почта, указанная пользователем при регистрации |

* 1. Выбор и обоснование комплекса программных средств

Программное средство – это объект, состоящий из программ, процедур, правил, а также сопутствующих им документации и данных, относящихся к функционированию системы обработки информации [].

Программное средство представляет собой конкретную информацию, объективно существующую как совокупность всех значимых с точки зрения ее представления свойств каждого из материальных объектов, содержащих в фиксированном виде эту информацию [].

* + 1. Выбор языка программирования

Язык программирования – формальный язык, предназначенный для представления программ []. В качестве языка написания основной логики программы был использован ХХХ. + описание достоинств

* + 1. Выбор операционной системы

Операционная система – совокупность системных программ, предназначенная для обеспечения определенного уровня эффективности системы обработки информации за счет автоматизированного управления ее работой и предоставляемого пользователю определенного набора услуг []. В качестве операционная системы была выбрана Windows 7.

Windows 7 – операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT [23].

Windows 7 поступила в продажу 22 октября 2009 года. С июля 2011 до марта 2017 Windows 7 занимала лидирующее положение по количеству пользователей в мире. По состоянию на январь 2019 доля Windows 7 составляет 18.8 % среди используемых в мире операционных систем для доступа к сети Интернет и занимает второе место в мире по популярности после Windows 10 [].

Компонентами пользовательского интерфейса Windows 7 являются окна с элементами управления, панель задач и контекстное меню, а основными устройствами ввода являются манипулятор, мышь, клавиатура, джойстик, (если в компьютере используется сенсорный экран, то средством ввода является дисплей, который преобразует нажатие или прикосновение к экрану в команду) [].

В Windows 7 впервые полностью представлена мультисенсорная технология. Функция Windows Touch доступна в выпусках Windows 7: Домашняя расширенная, Профессиональная и Максимальная [].

* + 1. Выбор среды программирования

Среда программирования – это набор инструментов, которые используются для преобразования символов в выполнимые вычисления [].

Обычно среда программирования включает:

1. редактор – средство для создания и изменения исходных файлов, которые содержат написанную на языке программирования программу;
2. компилятор – транслирует символы из исходного файла в объектный модуль, который содержит команды в машинном коде для конкретного компьютера;
3. компоновщик – собирает объектные файлы отдельных компонентов программы и разрешает внешние ссылки от одного компонента к другому, формируя исполняемый файл;
4. отладчик – это средство, которое дает возможность программисту управлять выполнением программы на уровне отдельных команд для диагностики ошибок [].

В качестве среды программирования выбрана ХХХ []. + ее достоинства

* + 1. Выбор СУБД

В качестве СУБД была выбрана… + ее достоинства

Заключение

В процессе выполнения выпускной работы была разработана автоматизированная система …, позволяющая ….

В первом разделе были приведены основные понятия и определения предметной области, здесь свое, приведены сравнительные характеристики систем-аналогов, на основании этого была сформулирована постановка задачи и основные требования к системе.

Во втором разделе была выбрана архитектура системы, разработана структура системы, спроектированы прототипы интерфейса пользователя, разработан информационно-логический проект системы по методологии UML, в состав которого вошли все канонические диаграммы, логическая модель данных, а также был выбран комплекс программных средств.

В третьем разделе был разработан и описан конечный интерфейс пользователя, выбран и обоснован комплекс технических средств, рассчитан объем занимаемой памяти, необходимый для работы системы, а также были установлены минимальные требования, предъявляемые к системе.

Разработанная автором система используется там-то и там-то. Имеется акт о внедрении разработанной системы в эксплуатацию (приложение В).

Результаты работы, приведенные в пояснительной записке, были представлены на международную конференцию «Математика. Компьютер. Образование – 2019» (г. Пущино, январь-февраль 2019 г.) [], докладывались 69 молодежной научной конференции Самарского университета, посвященной 85-летию со дня рождения первого космонавта Земли Ю.А. Гагарина (г. Самара, 5 апреля 2019 г.) (приложение Г), были представлены на международную научно-техническую конференцию «Перспективные информационные технологии. ПИТ-2019» (г. Самара, июнь 2019 г.) [31].

Имеется свидетельство о регистрации программы для ЭВМ в Федеральной службе по интеллектуальной собственности (приложение Д).

Список использованных источников

1 Покровская О.М. Совершенствование комплекса гигиенических мероприятий у пациентов с ортопедическими конструкциями на имплантатах. : дис. ... канд. мед. наук: 14.00.21: защищена 02.09.2008/ Покровская Ольга Михайловна. М., 2008. 115 с.: ил.

2 Богданов И.М., Зеленко Л.С., Филатова Н.В. Автоматизированная система расчета показателей основных стоматологических заболеваний у детского населения // Математика. Компьютер. Образование: труды XXIV межд. конф., 28 января – 2 февраля 2019 г., г. Пущино. URL: http://www.mce.su/rus/presentations/p333102/ (дата обращения: 02.02.2019).

3 Справочник по формулам зубов [Электронный ресурс]. URL: http://www.dental-revue.ru/index.php?page=15&artId=7 (дата обращения: 30.09.2018).

4 Современные системы оценки и регистрации кариеса зубов / Пастбин М.Ю., Горбатова М.А. и [др.]. Архангельск: Северный государственный медицинский университет, 2013. 7 с.

5 Методическое пособие по теме «Индексная оценка кариеса зубов и заболеваний пародонта» / Молоков В.Д., Доржиева З.В. и [др.]. Иркутск: Иркутский государственный медицинский университет, 2008. 23 с.

6 Официальный сайт Dental Software [Электронный ресурс]. URL: https://www.dentrix.com/ (дата обращения: 16.10.2018).

7 Официальный сайт Open Dental Software [Электронный ресурс]. URL: https://www.opendental.com (дата обращения: 18.10.2018).

8 Официальный сайт CS SoftDent [Электронный ресурс]. URL: https://www.carestreamdental.com/en-us/products/practice-management-software/cs-softdent/ (дата обращения: 20.10.2018).

9 Большой Российский энциклопедический словарь. М.: БРЭ, 2003. 341 с.

10 Буч, Г. Язык UML. Руководство пользователя [Текст] /Г. Буч, Д. Рамбо, А. Якобсон. 2-е изд.: Пер. с англ. Мухина Н. М.: ДМК Пресс, 2006. 496 с.: ил.

11 Диаграмма вариантов использования [Электронный ресурс] URL: https://www.intuit.ru/studies/courses/32/32/lecture/1004 (дата обращения: 15.11.2018).

12 Спецификация требований [Электронный ресурс] URL: https://www.intuit.ru/studies/courses/32/32/lecture/1006 (дата обращения: 9.10.2018).

13 Диаграмма классов [Электронный ресурс] // Студопедия: [сайт]. URL: https://studopedia.info/10-59449.html (дата обращения: 30.11.2018).

14 Основные понятия баз данных [Электронный ресурс]. URL: http://inf.susu.ac.ru/Klinachev/lc\_sga\_26.htm (дата обращения: 23.10.2018).

15 RFC 2898-2000. PKCS #5: Password-Based Cryptography Specification. Version 2.0 = Стандарт формирования ключа на основе пароля [Электронный ресурс]. URL: https://tools.ietf.org/html/rfc2898 (дата обращения: 15.11.2018).

16 ГОСТ 28806-90 Качество программных средств. Термины и определения. М., 1990. 12 с. (Издательство стандартов).

17 Официальный сайт SQLite [Электронный ресурс]. URL: https://www.sqlite.org/index.html (дата обращения: 01.07.2018).

18 ГОСТ 28397-89. Языки программирования. Термины и определения М., 1989. 8 с. (Издательство стандартов).

19 ISO/IEC 9899-2011. Programming languages C = Язык программирования C. Международный стандарт. США, Нью-Йорк: Американский национальный институт стандартов ANSI, 2012. 702 с.

20 ISO/IEC 14882-2014. Programming languages C++ = Язык программирования C++. Международный стандарт. Швейцария, Женева: Международная электротехническая комиссия IEC, 2014. 1375 с.

21 Основные принципы C++ [Электронный ресурс]. URL: https://en.cppreference.com/w/cpp/language/basic\_concepts (дата обращения: 02.07.2018).

22 ГОСТ 15971-90 Системы обработки информации. Термины и определения. М.: Издательство стандартов. 1991. 14 с.

23 Операционная система Windows 7 [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/Windows\_7 (дата обращения: 10.11.2018).

24 Знакомство с ОС Windows 7 [Электронный ресурс] URL: https://www.lessons-tva.info/edu/inf-win/win-1-2-1-1.html (дата обращения: 10.11.2018).

27 Пользовательский интерфейс [Электронный ресурс] URL: https://spravochnick.ru/informatika/arhitektura\_personalnogo\_kompyutera/polzovatelskiy\_interfeys/ (дата обращения: 15.11.2018).

28 Диаграммы реализации [Электронный ресурс] URL: http://www.maksakov-sa.ru/ModelUML/DiagrReal/index.html (дата обращения: 15.11.2018).

29 Диаграмма развёртывания [Электронный ресурс] URL: https://www.intuit.ru/studies/courses/32/32/lecture/1024 (дата обращения: 15.11.2018).

30 Физическая модель базы данных [Электронный ресурс] URL: https://studopedia.ru/6\_1898\_fizicheskaya-model-bazi-dannih.html (дата обращения: 25.10.2018).

31 Богданов И.М. Разработка автоматизированной системы расчета показателей основных стоматологических заболеваний у детского населения/ И.М. Богданов, Л.С. Зеленко, Н.В. Филатова // Перспективные информационные технологии (ПИТ-2019): сб. науч. тр. межд. научно-техн. конф.; [под ред. С.А. Прохорова]. Самара: Изд-во СНЦ РАН, 2019. С.238-342.