МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий   
Кафедра программной инженерии

Утверждаю

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_ Н.В. Пацей

подпись инициалы и фамилия

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021г.

**ЗАДАНИЕ**

**к курсовому проектированию**

**по дисциплине** "Объектно-ориентированное программирование"

|  |  |
| --- | --- |
| Специальность: 1-40 05 01 Информационные системы и технологии  Студент: Аникеенко Е. В. | Группа:\_\_2\_\_ |
| **Тема: Программная система «Общежитие»** | |

**1. Срок сдачи студентом законченной работы**: "20 мая 2021 г."

**2. Исходные данные к проекту:**

**2.1**. Функционально ПС поддерживает:

* Выполнять функции администратора:
  + Поддерживать работу c базой данных;
  + Выполнять поисковые запросы.
* Выполнять функции для пользователя:
  + Выполнять регистрацию и авторизацию;
  + Записываться на дежурство;

**2.2.** При выполнении курсового проекта необходимо использовать принципы и приемы ООП. Приложение разрабатывается под ОС Windows и представляет собой настольное приложение (desktop). Отображение, бизнес логика должны быть максимально независимы друг от друга для возможности расширения. Диаграммы вариантов использования, классов реализации задачи, взаимодействия разработать на основе UML. Язык разработки проекта – C#. Управление программой должно быть интуитивно понятным и удобным. При разработке использовать несколько наиболее подходящих шаблонов проектирования ПО.

**3. Содержание расчетно-пояснительной записки**

(перечень вопросов подлежащих разработке)

* Введение
* Постановка задачи и обзор литературы (алгоритмы решения, обзор прототипов, актуальность задачи)
* Проектирование архитектуры проекта (структура модулей, классов).
* Разработка функциональной модели и модели данных ПС (выполняемые функции)
* Тестирование
* Заключение
* Список используемых источников
* Приложения

**4. Форма представления выполненной курсовой работы:**

* + Теоретическая часть курсового проекта должны быть представлены в формате docx. Оформление записки должно быть согласно выданным правилам.
  + Листинги программы представляются частично в приложении.
  + Пояснительную записку, листинги, проект (инсталляцию проекта) необходимо загрузить диск, указанный преподавателем.

#### Календарный план

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта | Примечание |
| 1 | Введение | 19.02.2021 |  |
| 2 | Аналитический обзор литературы по теме проекта. Изучение требований, определение вариантов использования | 12.03.2021 |  |
| 3 | Анализ и проектирование архитектуры приложения (построение диаграмм, проектирование бизнес-слоя, представления и данных) | 26.03.2021 |  |
| 4 | Проектирование структуры базы данных. Разработка дизайна пользовательского интерфейса | 2.04.2021 |  |
| 5 | Кодирование программного средства | 23.04.2021 |  |
| 6 | Тестирования и отладка программного средства | 30.04.2021 |  |
| 7 | Оформление пояснительной записки | 7.05.2021 |  |
| 9 | Сдача проекта | 20.05.2021 |  |

**5. Дата выдачи задания \_\_\_\_**12.02.2021**\_\_\_\_**

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *А. В. Годун*

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата и подпись студента)

**Оглавление**

[ВВЕДЕНИЕ 4](#_Toc72494516)

[1. АНАЛИТИЧЕСКИЙ ОБЗОР 5](#_Toc72494517)

[1.1 Preno management system 6](#_Toc72494518)

[1.2 The Booking Factory 10](#_Toc72494519)

[1.3 Sirvoy 12](#_Toc72494520)

[1.4 YuKoSoft Гостиница 14](#_Toc72494521)

[2. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ПРОЕКТА 18](#_Toc72494522)

[2.1 Архитектура системы 18](#_Toc72494523)

[2.2 Проектирование логической структуры базы данных 18](#_Toc72494524)

[2.3 Проектирование структуры окон 19](#_Toc72494525)

[2.4 Ресурсы проекта 21](#_Toc72494526)

[2.5 Пример варианта использования 22](#_Toc72494527)

[2.6 Структура проекта 22](#_Toc72494528)

[3. РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ МОДЕЛИ И МОДЕЛИ ПС 24](#_Toc72494529)

[3.1 Взаимодействие с базой данных 24](#_Toc72494530)

[3.2 DataGrid 24](#_Toc72494531)

[3.3 Валидация 25](#_Toc72494532)

[3.4 Схема и описание авторизации 25](#_Toc72494533)

[3.5 Использованные готовые решения 26](#_Toc72494534)

[3.6 Перемещение по страницам 26](#_Toc72494535)

[4. ТЕСТИРОВАНИЕ 27](#_Toc72494536)

[4.1 Тестирование валидации 27](#_Toc72494537)

[4.2 Тестирование исключительных ситуаций 27](#_Toc72494538)

[4.3 Использование приложения обычным пользователем 28](#_Toc72494539)

[4.4 Использование приложения администратором 30](#_Toc72494540)

[ЗАКЛЮЧЕНИЕ 32](#_Toc72494541)

[Список используемых источников 33](#_Toc72494542)

[ПРИЛОЖЕНИЕ А 34](#_Toc72494543)

[ПРИЛОЖЕНИЕ Б 35](#_Toc72494544)

[ПРИЛОЖЕНИЕ В 36](#_Toc72494545)

[ПРИЛОЖЕНИЕ Г 37](#_Toc72494546)

[ПРИЛОЖЕНИЕ Д 38](#_Toc72494547)

[ПРИЛОЖЕНИЕ Е 39](#_Toc72494548)

ВВЕДЕНИЕ

Проблема контроля общежития довольно обширна. Поскольку, этой проблемой занимаются всего лишь несколько человек, при этом работая с огромными списками, а сами жильцы не против помочь в данной сфере и не ходить к старосте этажа для того, чтобы выполнить банальное действие и записаться на дежурство, а просто отметиться в каком-либо электронном списке. Также зачастую возникает желание узнать некую информацию о самом общежитии, либо о своих соседях и иных жильцах. Для всех этих целей и была поставлена задача создания удобного приложения для администрирования общежития.

Целью данной работы является создание менеджера, работающего с информацией, необходимой для удобного функционирования общажного аппарата. Данный менеджер предполагает возможность регистрации пользователя и разделение прав администратора и обычного жильца.

Это программное средство ставит перед собой задачу по упрощению и повышению удобства:

* управления списком жильца;
* управления дежурствами;
* управления пользователями программы;
* поиск по спискам жильцов;
* использования программы за счет графического интерфейса.

Таким образом, разработка приложения позволит пользователю администрировать общежитие и проводить сбор необходимой информации.

Структура приложения: приложение, написанное на С#. Оно устанавливается на ПК и представляет собой интерфейс.

1. АНАЛИТИЧЕСКИЙ ОБЗОР

В данном разделе я хочу изучить существующие веб и десктоп приложения, которые позволяют управлять общежитием, а также в сравнение будут включены существующие порталы для бронирования комнат, потому как функционал несколько схож с планируемым приложением.

Для того, чтобы рассмотреть аналоги, необходимо продумать, по каким критериям я буду их оценивать. В данном случае исходники приложений недоступны, поэтому для оценки будут рассматриваться только юзабилити характеристики. В качестве критериев будут выступать следующие показатели:

* контент;
* функционал;
* кроссбраузерность или кроссплатформенность;
* навигация;
* дизайн;
* скорость работы.

Теперь поговорим немного про сами характеристики и почему именно они были выбраны в качестве оценочных. Начнем с характеристики под названием «контент», данная характеристика позволяет оценить информацию и тексты, которые размещаются на сайте – основной смысл. Тексты должны быть максимально читабельны и понятны. Если у пользователя возникнут трудности в чтении, понимании, то скорее всего он покинет сайт раньше, чем вы бы этого хотели.

Следующая характеристика «функционал», которая позволяет оценить наличие ошибок, недочётов, а также лишние элементы, которые будут всегда всплывать. Данная характеристика является, пожалуй, одной из самых главных, т. к. она требует, чтобы разработчик своевременной реагировал на критику, замечал и быстрее избавлялся от ненужных функций и добавлял те, что требует пользователь, чтобы не потерять клиента.

За функционалом следует немаловажная характеристика «кроссбраузерность» или «кроссплатформенность», данная характеристика влияет не столько на само содержание приложения, сколько на возможность им воспользоваться потенциальному потребителю. Поэтому приложение должно отображаться и работать во всех популярных браузерах и на всех планируемых платформах одинаково хорошо. Проверить можно самостоятельно, открыв сайт во всех браузерах или установив приложения на целевые платформы и сравнив опыт от взаимодействия.

Характеристикой, которая позволяет оценить, насколько быстро и просто пользователь сможет добираться до конкретных опорных точек функционала и выполнять желаемые действия, является «навигация». Навигация должна быть качественно продумана, чтобы пользователь всегда мог перейти на главную приложения или вернуться на шаг назад, чтобы пользователь имел возможность перейти на смежные страницы, что позволит сэкономить время и повысит эмоциональный настрой от использования.

Одной из характеристик, которая часто является решающей для конечного пользователя, служит «дизайн». Дизайн – это первое, что видит пользователь после открытия приложения, наличие нагромождения, которое повышает время доступа к контенту, чрезмерное количество информации и всплывающих окон, неприятные и непритягательные цвета, а также цвета, которые не соответствуют действию, которое можно было бы ожидать, примером этого являются объемные кнопки, понятные и общепринятые цвета ссылок, а также изменение элемента управления в зависимости от действия, например, при наведении на кнопку меняется стиль курсора, а также сам стиль кнопки, во время нажатия кнопка также должна менять свой стиль, чтобы было понятно, чтобы было произведено действие.

И последняя по очереди, но не по значимости, идет характеристика под названием «скорость работы». Данная характеристика позволяет оценить время, которое затратит пользователь на выполнения тех или иных действий, а также скорость обработки данных. Эта характеристика часто является решающей в случае, когда присутствуют аналоги, возможно уступающие в некоторых других характеристиках, но позволяют оперативно и без лагов выполнить основную работу.

* 1. Preno management system

В качестве первого аналога выступает веб приложения для бронирования комнат. Данное приложение предоставляет бесплатную версию на 30 дней, которой я могу воспользоваться для составления оценки.

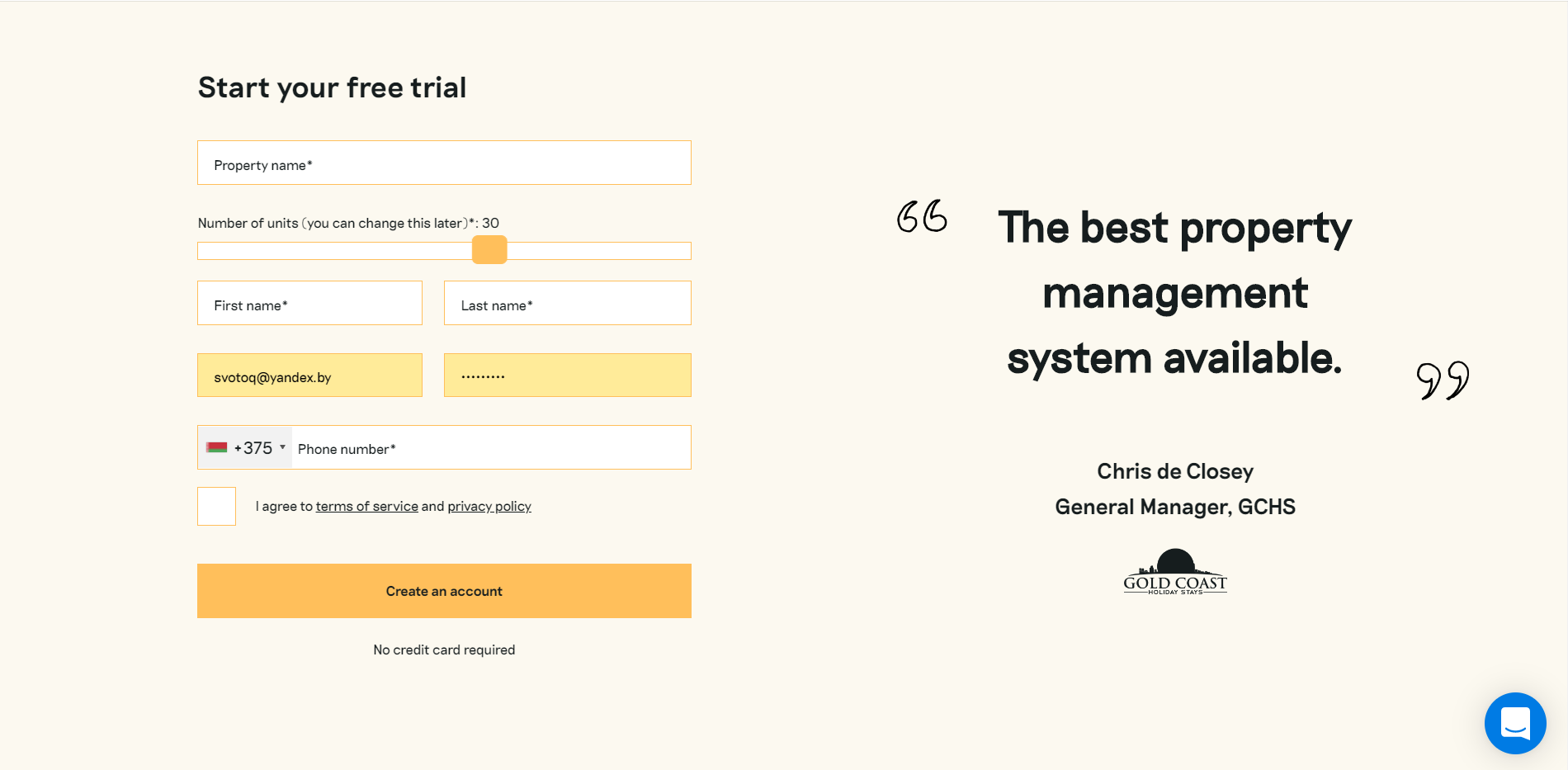


Рисунок 1.1 – Оформление демо версии

После оформления демо версии, приложение перенаправляет на главную страницу приложения без необходимости подтверждать что-либо по почте или номеру телефона.

Следующим предстает главная страница с большой таблицей, с помощью которой и происходит бронирование, то есть это основной функционал, и он вынесен на главную.

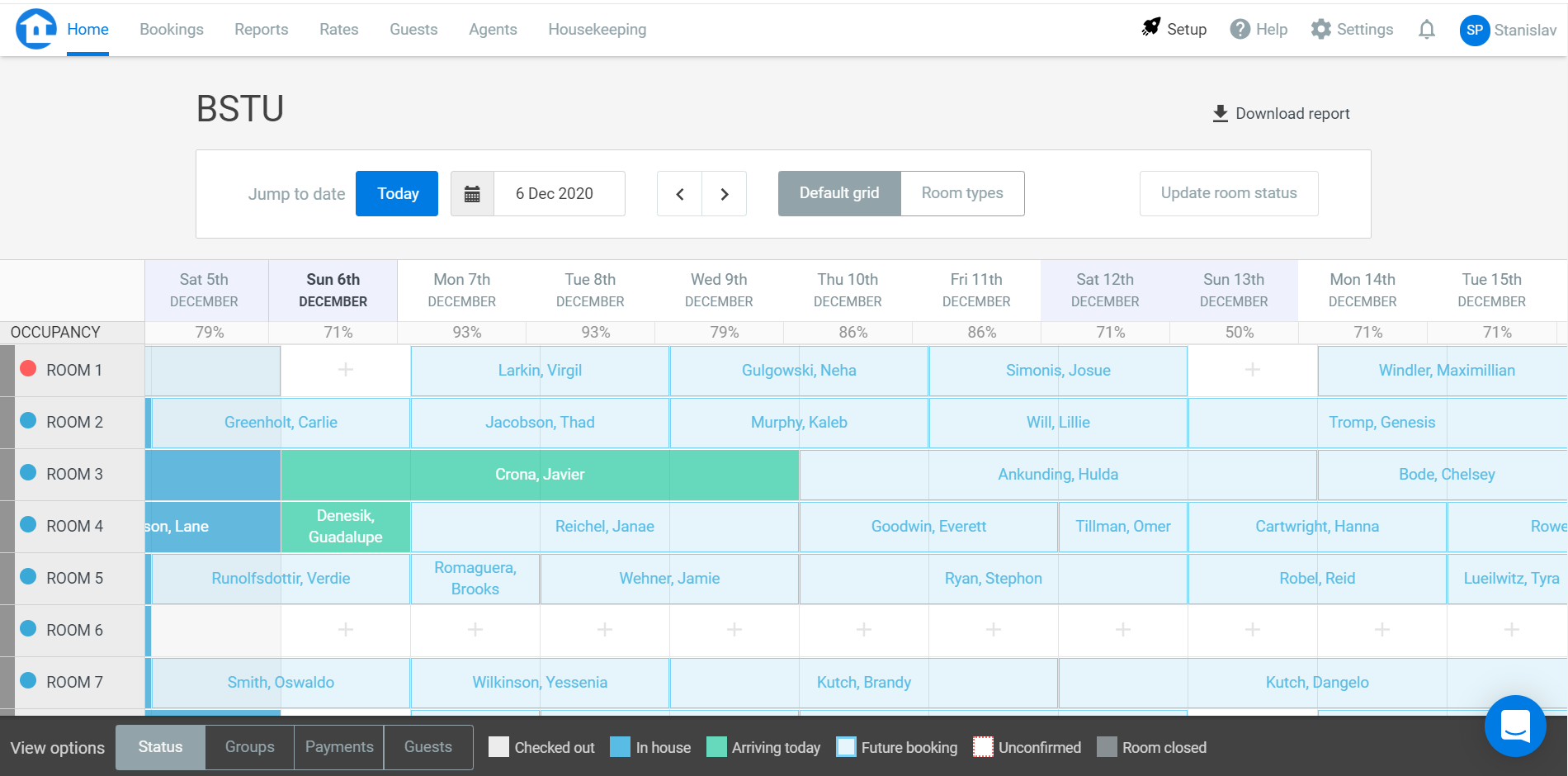


Рисунок 1.2 – Главная страница приложения

При первом входе в приложение легко растеряться из-за большего нагромождения различных цвет и элементов управления, но после нескольких минут тщательного изучения весь функционал становится понятным.

Следует начать с того, что главная страница обладает основным функционалом, простой навигацией по датам и простой системой бронирования, а также доступна фильтрация по типам комнат. Чтобы проще было ориентироваться по таблице, внизу представлена сноска, которая показывает информацию о том, за что каждый цвет отвечает. Несомненным плюсом является кнопка скачивания

Следующим идет меню в самом верху, чтобы можно было переходить на основные вкладки и просматривать информацию о забронированных комнатах, уведомлениях, гостях, которые осуществили бронирование (почту и другие персональные данные), также можно выбрать сервисы, на которых размещены объявления о свободных комнатах, например, «Booking» и «Agoda», для настройки можно указать параметры, такие как такса, которую взымают сервисы за свои услуги и взымается ли эта плата с гостя.

Последней вкладкой является «Housekeeping», которая отвечает за информацию о состоянии жилья, данная вкладка является полезной, т. к. позволяет своевременно отслеживать проблемные комнаты, и кто в них жил, а также своевременно убираться в помещении.

Приложение не заканчивается только на основном функционале, следует рассмотреть наличие документации или возможности обратной связи с разработчиками.

Для этого в приложении была обнаружена вкладка «Help», которая отвечает за возможность задать вопрос или найти ответ среди существующих.

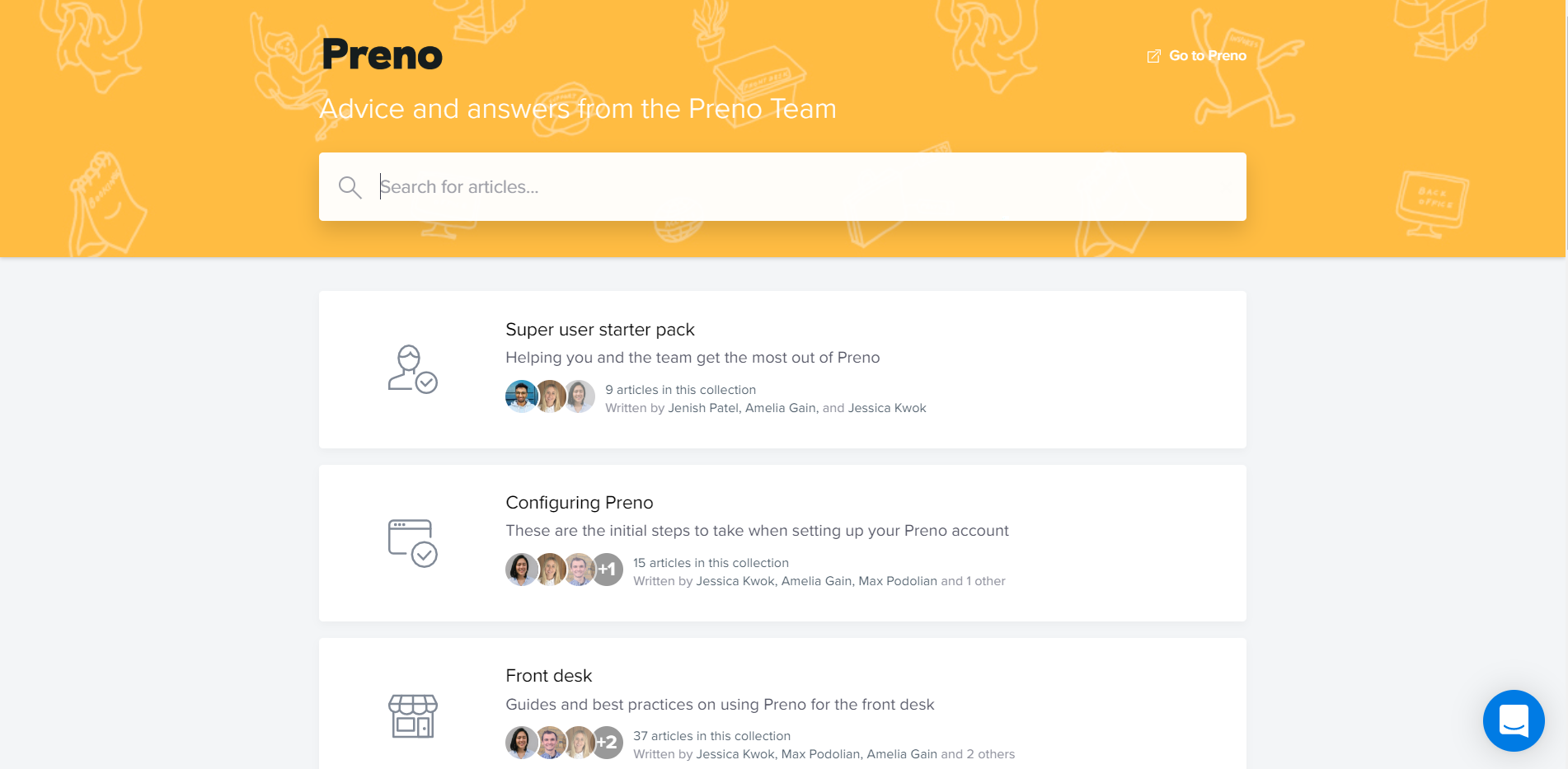


Рисунок 1.3 – Руководство приложения

Несомненным плюсом является то, что помимо документации, приложение предоставляет советы с помощью которых можно повысить эффективность работы и раскрыть все возможности приложения. Отсутствие русского языка или возможности изменить его на любой другой кроме английского это несомненный минус, который отсекает большую часть потенциальных клиентов.

После просмотра всего сайта можно провести оценку и подвести итоги:

контент на сайте является исключительно рабочим, если не считать главную на которой и происходило оформление демоверсии приложения. Приложение не содержит орфографические или пунктуационные ошибки, название для элементов управления подобраны правильно, снизить бал можно за отсутствие русского языка. Документация является качественной и продуманной, обратная связь возможна с любой из страниц. За контент можно поставить 4 из 5 баллов;

функционал приложения удовлетворяет всем требованием, наличие графиков и диаграмм, возможность автоматизировать некоторые процессы по отправлению писем, и своеобразная интеграция с популярными сервисами для бронирования позволяет оценить приложение в 5 из 5 баллов.

кроссбраузерность, приложение работает на всех современных браузерах исключая устаревший Internet Explorer. На мобильных устройствах сайт выглядит неудобным для использования, минимальный размер устройства с которого комфортно пользоваться сайтом это 11 домовый планшет, поэтому сайт заслуживает оценку 4 из 5;

навигация в приложении достаточно удобная и позволяет с любой вкладки оказаться на любой другой, каждая страничка приложения выполняет только единственную функцию что является несомненным плюсом, поэтому за навигацию приложение заслуживает оценку 5 из 5;

дизайн в приложении слишком простой, цвета подобраны сильно блеклые и серые, при постоянном использовании можно потерять интерес к жизни, из плюсов стоит отметить шрифт и выдержанную цветовую палитру для всех страниц, в остальном же, дизайн слишком простой и не заслуживает внимания, хотя и не сбивает с толку, поэтому 3 из 5;

скорость работы, приложение не может удивить моментальной скоростью работы, потому как находится в облаке и некоторые аспекты зависят от скорости интернета, но даже при хорошем интернет-соединении многие функции выполняются долго и могли бы быть более оптимизированными, но при этом не настолько долго, чтобы начать раздражаться, поэтому оценка 3 из 5.

Подводя итоги по данному приложению, он набрал 24 балла из 30 возможных, что является хорошим результатом.

* 1. The Booking Factory

Следящее приложение будет также относиться к веб и также предоставляет демоверсию на 30 дней, но в данном случае требуется подтверждение по почте, но после этого предоставляется беспрепятственные доступ к приложению.

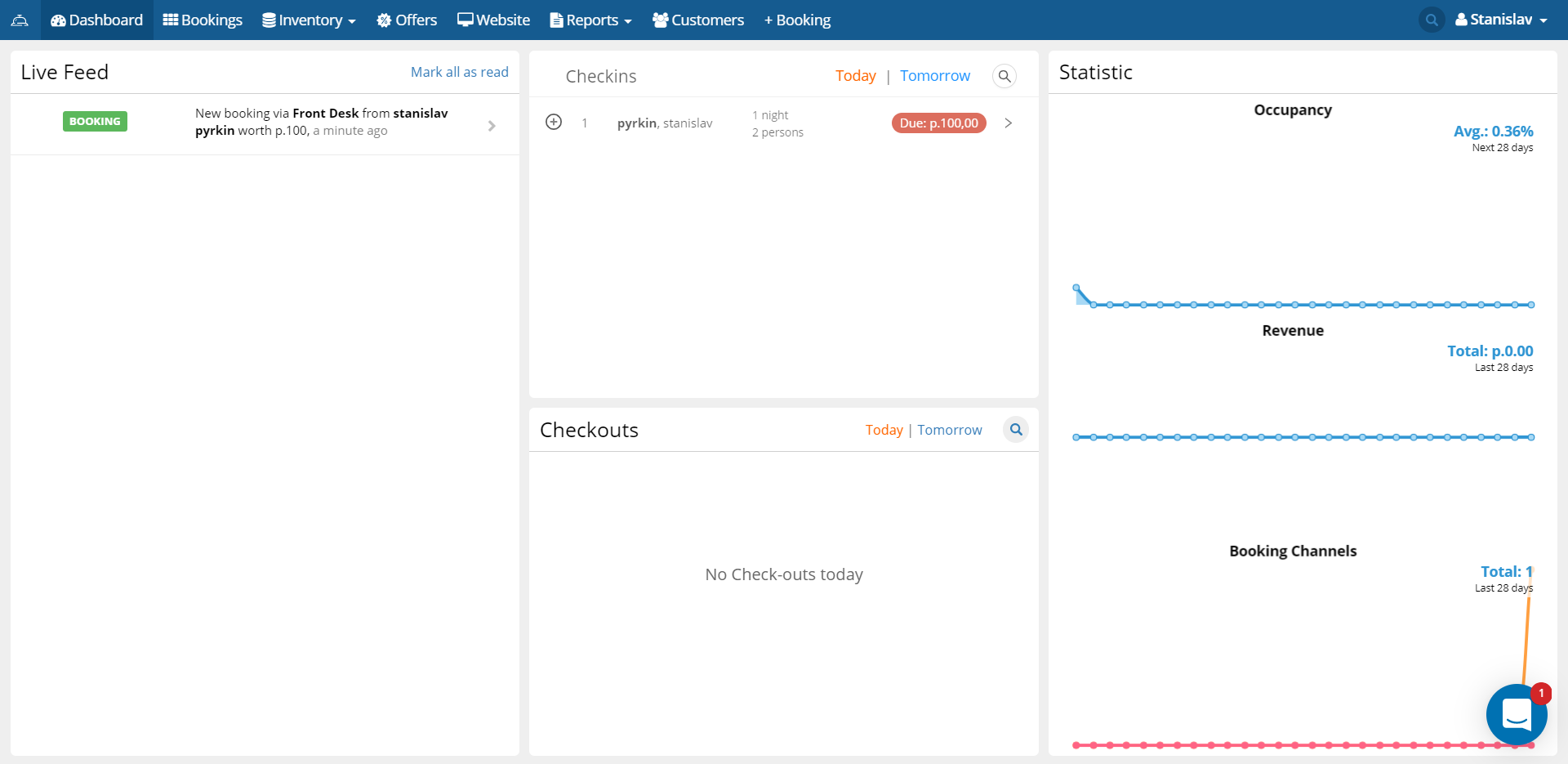


Рисунок 1.4 – Главная страница приложения

Приложение относится к категории бронирования комнат. При входе на главную нас встречает интерфейс, который сразу содержит статистику, новостную ленту, которая отражает последние действия, а также списки забронированных и уже освободившихся комнат.

При первом входе в приложение я слегка растерялся, пришлось потратить некоторое время, чтобы понять функционал, но он достаточно схож с предыдущим приложением. Начнем с оценки контента, в общем, он достаточно отражает и вполне советует ожиданиям, но есть некоторые моменты, вроде «Inventory» или другие неочевидные названия, также отсутствует подсвечивание обязательных полей, поэтому, чтобы забронировать комнату или создать акцию, ушло много времени, чтобы понять почему кнопка «Save» недоступна, документация точно такая же как и в предыдущем приложении, доступен только английский язык, поэтому я оцениваю в 3 из 5 баллов.

Функционал приложения очень порадовал, здесь присутствуют дополнительные функции, которых не было в предыдущем приложении, вроде создания акций, добавления в заказ обеда и т. п.

Функционал создания сайта очень гибкий и позволяет использовать не только встроенную систему бронирования, но и собственную. При этом для создания сайта присутствует более 5 шаблонов с помощью которых можно изменить внешний вид сайта. Создание сайта очень простое, необходимо ввести определенный текст в те места, где он должен быть и загрузить свои картинки. Функционал выше всяких похвал, однозначно 5 из 5 баллов.

Настала очередь кроссбраузерности и кроссплатформенности, основное приложения для управления бронированием не оптимизировано под мобильные устройства, а сайт, который можно создать из шаблона вполне удобен при использовании телефона. Сайт одинаково выглядит во всех браузерах, но отсутствие удобного интерфейса при использовании телефона или планшета отнимает два балла, поэтому 3 из 5 возможных баллов.

Навигация в приложении качественная и позволяет вернуться или перейти на нужную вкладку достаточно быстро и понятно, весь функционал доступен с главной панели, которая видна на всех страницах, навигация заслуживает оценки 5 из 5 баллов.

Дизайн в приложении более интересный, нежели в предыдущем, но оно все также использует не очень яркие и насыщенные цвета, а также их сочетания, что свидетельствует о незамысловатом дизайне, а также некоторые элементы выполнены недостаточно корректно, что затрудняет использование, поэтому 2 из 5 баллов.

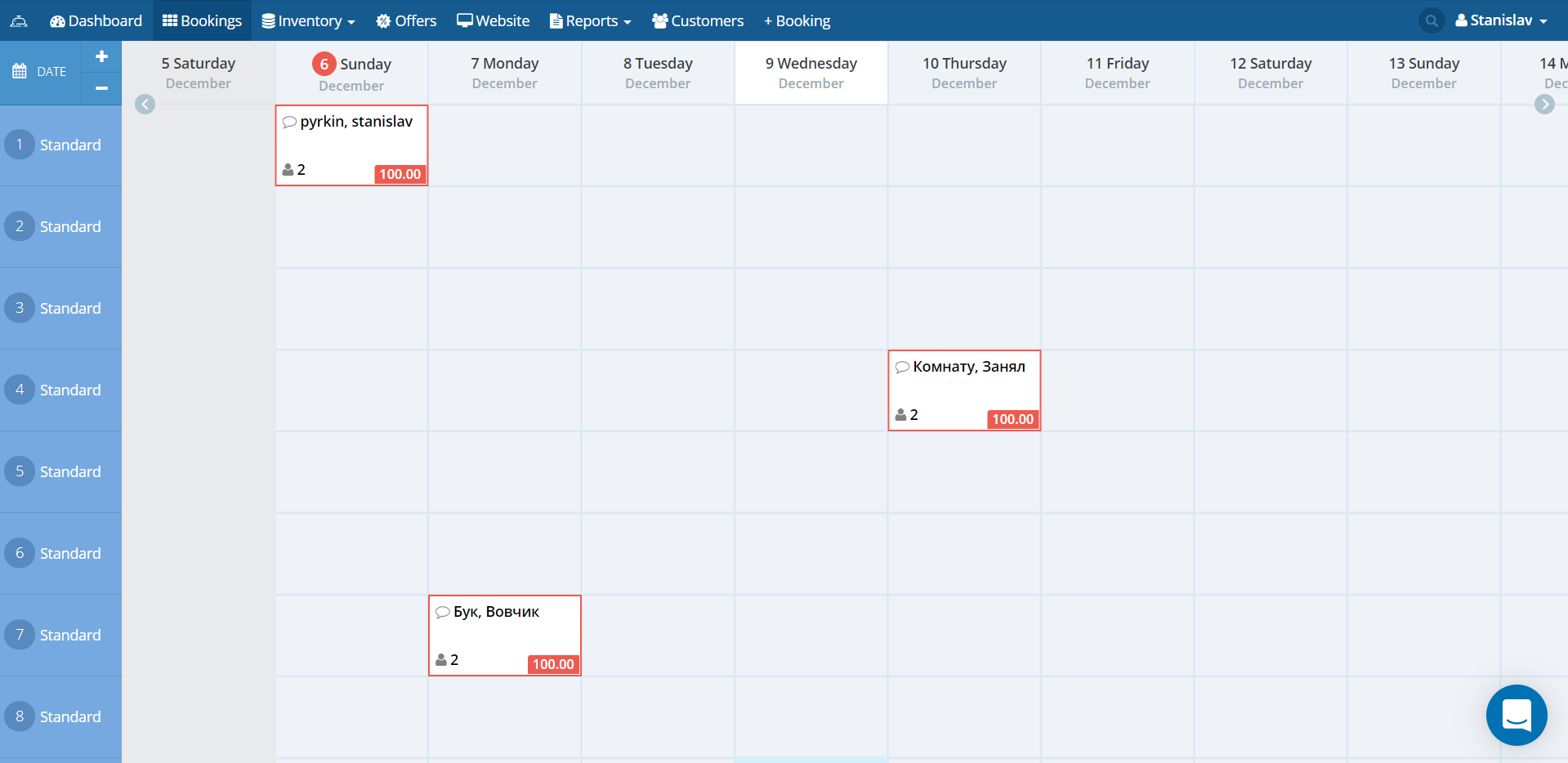


Рисунок 1.5 – Страница бронирования

Страница бронирования визуально отличается и находится на отдельной странице в отличии от предыдущего приложения, что является несомненным плюсом, а также занимает всю доступную площадь и таблица более удобная для использования.

Скорость работы приложения и собственного сайта, который можно создать с помощью конструктора достаточная высокая, но при этом некоторые неочевидные вещи подтормаживают, поэтому можно оценить в 4.5 из 5 баллов.

Итоговая оценка приложения составила 22,5 балла из 30 возможных, что немного хуже предыдущего приложения.

* 1. Sirvoy

Следующее приложение предоставляет более обширные возможности и настройки по бронированию. Данное приложение предоставляет тестовый период на 14 дней, что значительно меньше, чем у его конкурентов. Для ознакомления с возможностями этого времени вполне достаточно, но недостаточно будет проверить его на практике и целесообразность оформить подписку.

При входе в приложение нас встречает главная страничка, на которой можно сразу увидеть статистику и выполнить поиск по параметрам.

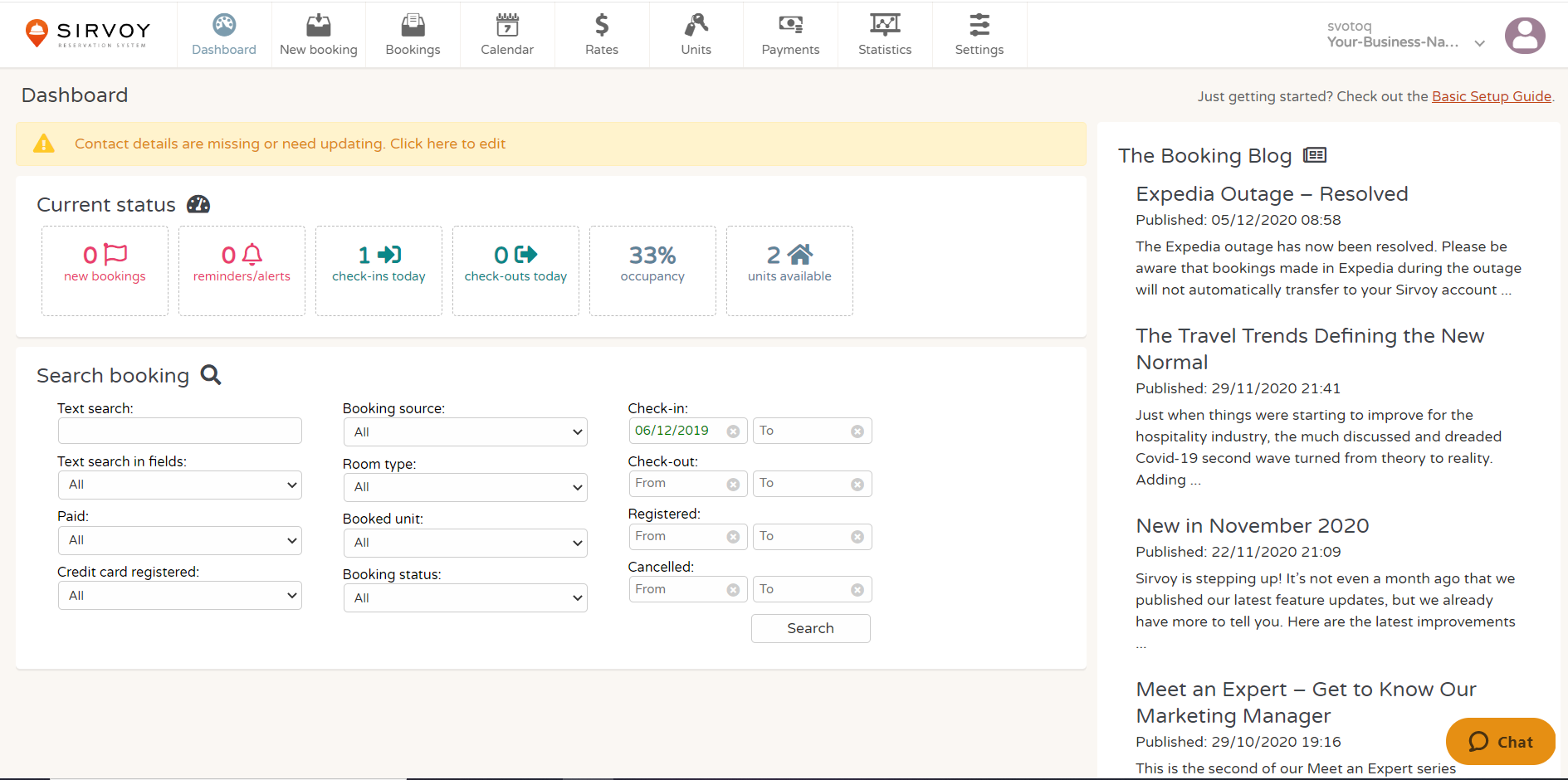


Рисунок 1.6 – Главная приложения

Сразу перейдем к оцениванию приложения, начнем с контента. В приложении очень много текста и параметров, а также присутствует новостная лента, которую нельзя отключить, что является не очень удобным и полезным для некоторых пользователей. Следующее на что хотелось бы обратить внимание это инструкция, она очень краткая и не очень понятная, поэтому у меня возникли большие сложности во время первой настройки и выполнения бронирования, за контент однозначно ставлю 2 из 5 баллов.

Функционал в приложение как это не странно достаточно обширный со множеством критериев и параметров, возможно многофункциональность – это хорошее решение, но без возможности отключать ненужные функции, чтобы сосредоточится только на необходимых является минусом. Основные функции реализованы в полной мере, поэтому 4 из 5 баллов.

Кроссбраузерность в приложении реализована в полной мере. Кроссплатформенность реализована не в полной мере, приложение работает корректно на мобильных устройствах, но из-за большего числа полей и нагромождения, пользоваться на мобильном телефоне через чур неудобно, а вот с планшетом приложение совместимо в полной мере, поэтому итоговая оценка 4 из 5 баллов.

Навигация в приложении не совсем интуитивно понятная и не всегда попадаешь туда, куда нужно, а также не все действия получается выполнить сразу т. к. нет последовательности для выполнения действий, которые оказываются обязательными, за навигацию ставлю 3 из 5 баллов.

Дизайн выполнен в серо-белых тонах, что создает дискомфорт при использовании, зато уведомления на таком фоне хорошо заметны. Стоит отметить наличие валидации полей и их подсветку, что позволяет очень быстро сориентироваться. В остальном дизайн на твердую 3 из 5 баллов, пользоваться можно, но все очень тускло.

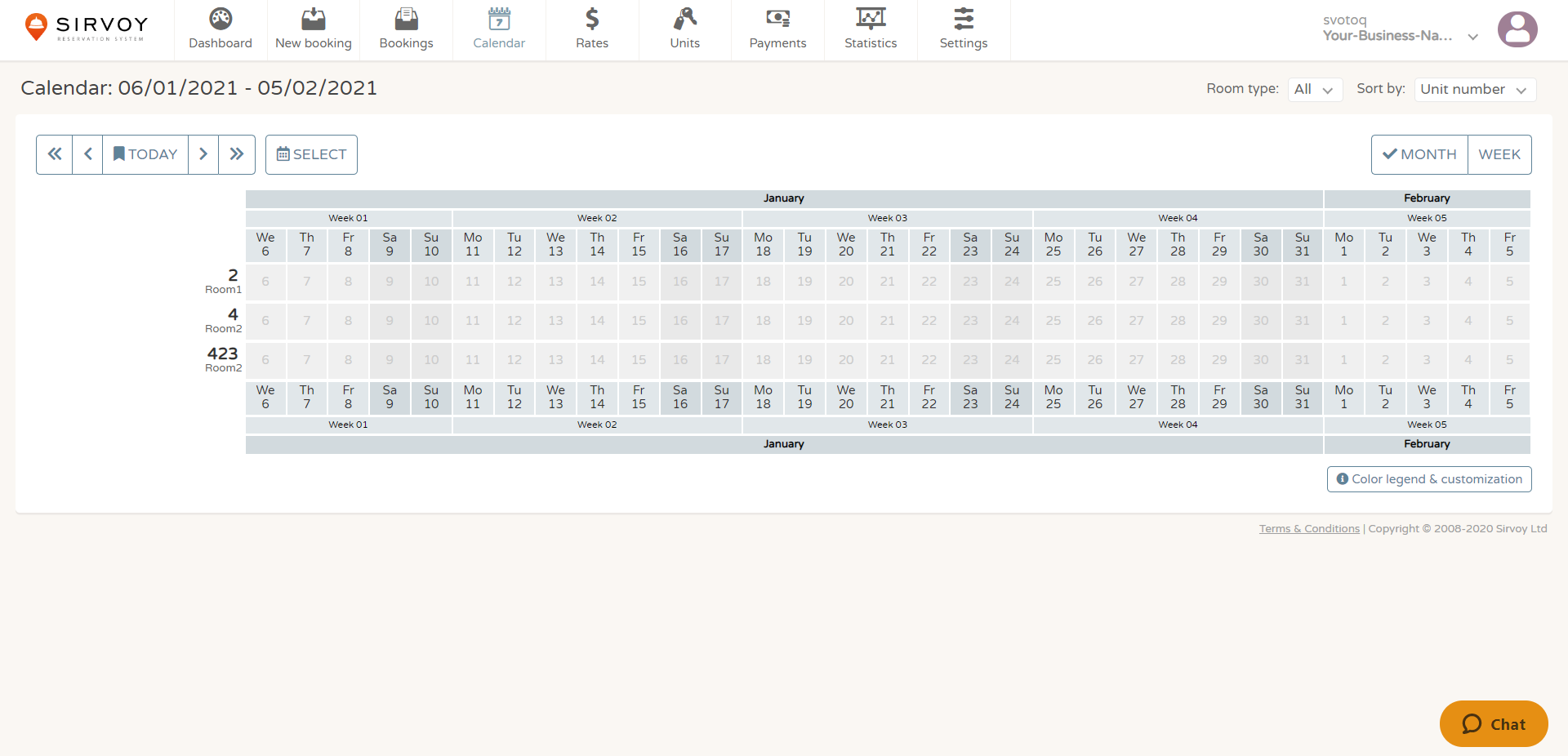


Рисунок 1.7 – Календарь бронирования

Скорость работы приложения приемлемая и ничего не подвисает на незначительных действиях, скорость поиска достаточно высокая, возможно при наличии большой базы данных скорость будет ниже, но в сравнении с конкурентами, данное приложение лидирует по скорости, поэтому твердые 5 баллов.

Подводя итоги по данному приложению, хочется отметить, что оно отличается от предыдущих визуально и имеет значительно больший функционал, хотя и слегка запутанный, конечная оценка 21 балл из 30 возможных.

* 1. YuKoSoft Гостиница

Последним идет приложение для персональных компьютеров. Оно предоставляет бесплатную полную версию до 60 запусков приложения. После 60 запусков действуют некоторые ограничения.

Главное окно состоит из трех областей – это навигация, вкладки, которые были выбраны через навигацию, и окно заметок. Удобным является то, что можно открыть несколько окон и быстро переключатся между ними, что позволит оставить только необходимый функционал, возможность сворачивать подпункты меню – это также хорошее решение.

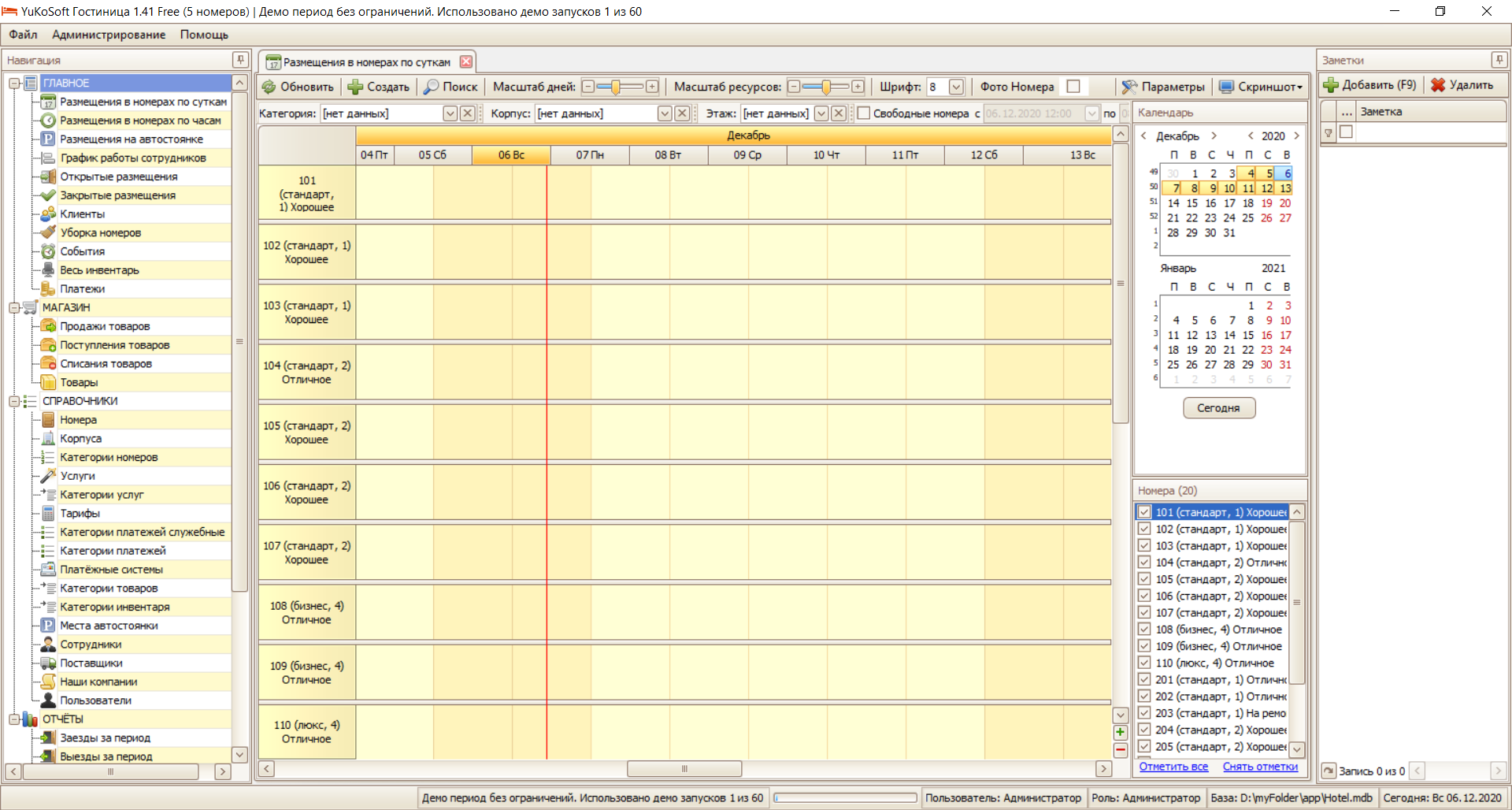


Рисунок 1.8 – Главная приложения

Контент приложения полностью на русском языке без возможности изменения его. Все пункты приложения описаны верно, документации для приложения не предусмотрено, но интерфейс достаточно прост и понятен, поэтому оценка составляет 4 из 5 баллов.

Функционал приложения позволяет сделать все необходимые для администрирования действия, возможность настраивать отображение необходимого функционала и наличие вкладок, возможность экспортировать и импортировать данные и графики, отслеживание графиков сотрудников, учет времени прибытия, использования автостоянки, оценка 5 из 5 баллов.

Кроссплатформенность у данного приложения плохая, т. к. оно доступно только на компьютерах, которые работают на операционной системе Windows. Оценка приложению 2 из 5 баллов.

Навигация в приложении очень простая и доступная со всех окон, возможность сворачивать ненужные пункты, вкладки отображают ровно то, что ожидается от приложения, поэтому за навигацию ставлю 5 баллов.

Дизайн у приложения очень старый и простой, цвета подобраны неудачно, многие иконки слишком мелкие, а другие слишком большие, некоторые элементы интерфейса отображаются непонятно для пользователя, но пользоваться приложением, в целом, удобно и приятно, поэтому оценка 3 из 5 баллов.

Приложение работает с задержками и лагами, некоторые действия выполняются непозволительно долго, открытие вкладок происходит резко, в ходе использования приложение несколько раз зависало, не уведомляя пользователя, что происходят какие-то операции, оценка 2 из 5 баллов.

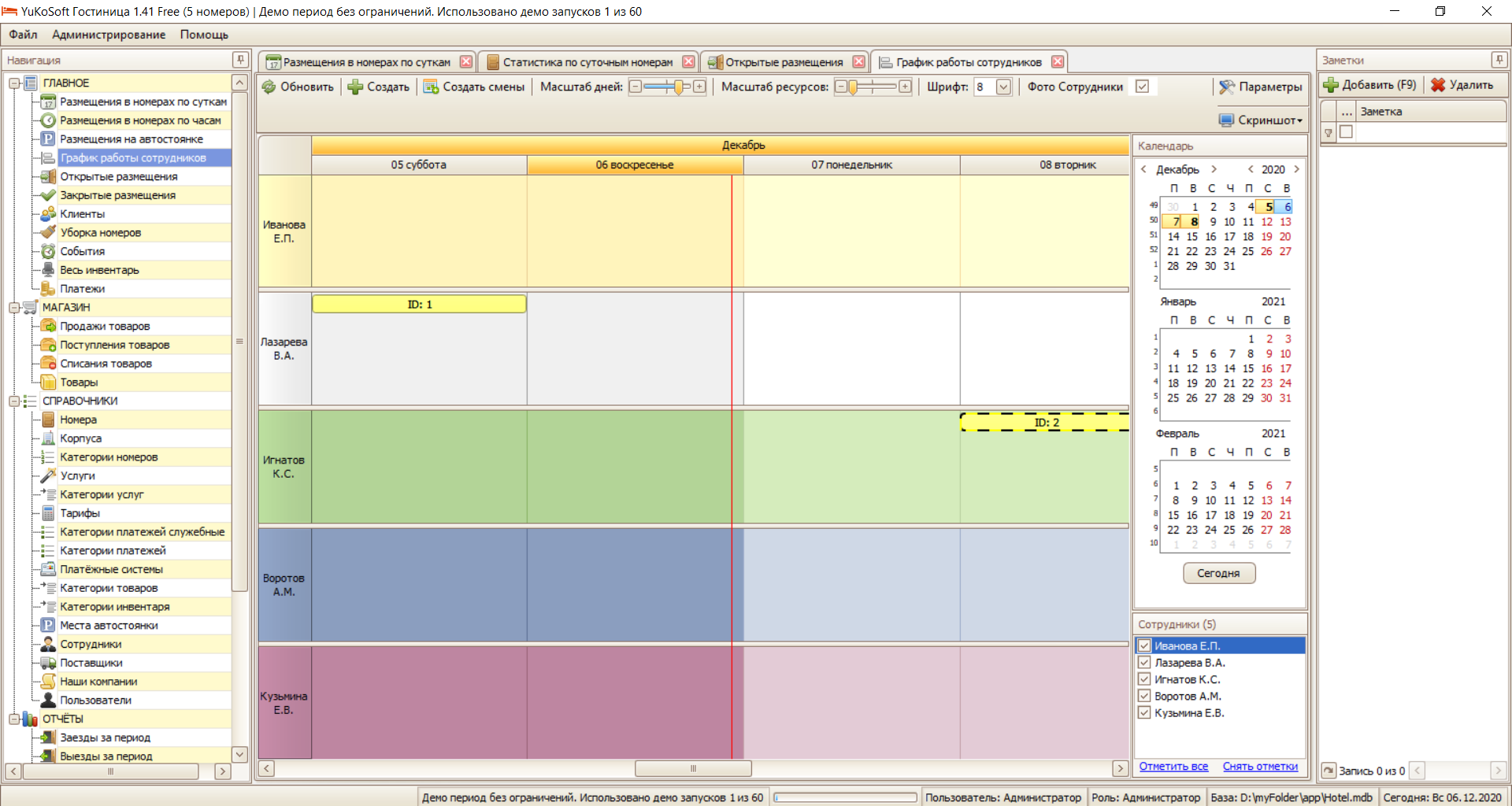


Рисунок 1.9 – График работы сотрудников

Суммирование оценок всех характеристик приложения приносят ему 21 балл из 30 возможных.

После оценки всех приложений можно сделать вывод, что многие имеют схожий интерфейс и концепцию, но каждое приложение реализовало его по-своему. В ходе изучения аналогов, больше всего не понравились приложения, которые были не дружелюбны с пользователем, то есть не содержали подсветки полей, которые являются обязательными, не уведомляли пользователей о наличии ошибок. Приложения, которые обладали излишним функционалом, но при этом не позволяли настраивать видимость также вызывали негативную реакцию. В разрабатываемом приложении следует обратить внимание на то, чтобы оно работало ожидаемо быстро в тех действиях, которые не должны вызывать задержек, не зависало без уведомления пользователя, что идет загрузка чего-либо и было дружелюбным. Для оформления приложения стоит подобрать яркую и в то же время мягкую цветовую палитру, с которой можно работать несколько часов и не уставать. Следует реализовать возможность поэтапного выполнения действий, что позволяло фокусироваться на единственной задачи, избегать расфокуса внимания и было дружелюбно для нового пользователя, который еще не был знаком с данной или похожими системами ранее, но стоит учесть общий концепт приложений, чтобы пользовали, которые имеют опыт использования могли быстрее адаптироваться.

2. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ПРОЕКТА

2.1 Архитектура системы

Для программного средства в курсовом проекте будет использоваться монолитная архитектура.

Монолитное приложение (монолит) представляет собой приложение, доставляемое через единое развертывание. Таким является приложение, доставленное в виде одной WAR или приложение Node с одной точкой входа.

Большим преимуществом монолита является то, что его легче реализовать. В монолитной архитектуре можно быстро начать реализовывать бизнес-логику, вместо того чтобы тратить время на размышления о межпроцессном взаимодействие.

Еще одна вещь — это сквозные (E2E) тесты. В монолитной архитектуре их легче выполнить.

Говоря об операциях, важно сказать, что монолит прост в развертывании и легко масштабируется. Для развертывания можно использовать скрипт, загружающий модуль и запускающий приложение. Масштабирование достигается путем размещения Loadbalancer перед несколькими экземплярами приложения.

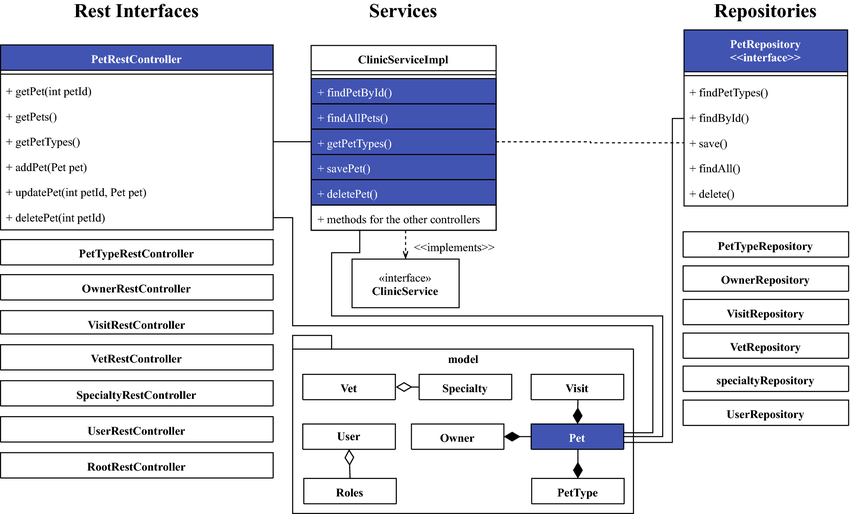


Рисунок 2.1 – Пример UML диаграммы для монолитной архитектуры

2.2 Проектирование логической структуры базы данных

При разработке программного средства, вся информация, которая заполняется пользователями и регистрируется программой должна храниться в определенных таблицах базы данных.

Для реализации поставленной задачи была создана база данных. Для её создания использовались система управления реляционными базами данных Microsoft SQL Server 2012.

База данных – совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимо от прикладных программ.

Система управления базами данных – совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения взаимодействия ее с прикладными программами.

Таблица Users используется для хранения данных пользователя. Включает в себя 5 столбцов: Id, Login, Password, Nickname и Status. Столбец Id является ключевым и уникальным, в нем хранится идентификатор пользователя. Столбец Login содержит логин, а столбец Password — захешированный пароль. В столбце Status содержится информация о статусе пользователя, он может содержать один из двух возможных типов:

* Admin;
* User.

Таблица Tenants используется для хранения данных о жильцах общежития. Включает в себя 8 столбцов: Id, Name, LastName, Patronymic, Sex, Room, Course, Group. Столбец Id является идентификатором (его предназначение описано выше). В свою очередь Name, LastName и Patronymic представляют собой имя, фамилию и отчество соответственно. Также из личной информации хранится курс (Course) и группа (Group) жильца. Для удобства сортировки и выборки был добавлен пол жильца (Sex).

Таблица Duties представляет собой расписание дежурств на все месяцы, когда использовалась программа и содержит 3 столбца: Id, TimeOfDuty, Orderly. Столбец Id является идентификатором (описание представлено выше). TimeOfDuty представляет собой время дежурства, которое заполняется раз в месяц при наступлении первого числа. Столбец Orderly заполняется пользователем, который желает дежурить во время TimeOfDuty или же администратор может записать туда любого человека.

Логическая структура базы данных представлена в приложении А.

## 2.3 Проектирование структуры окон

Приложение включает в себя 9 окон. Связи между ними представлены на рисунке 2.1. Во всех окнах есть возможность возврата к предыдущему окну, а также выхода из данного пользователя.

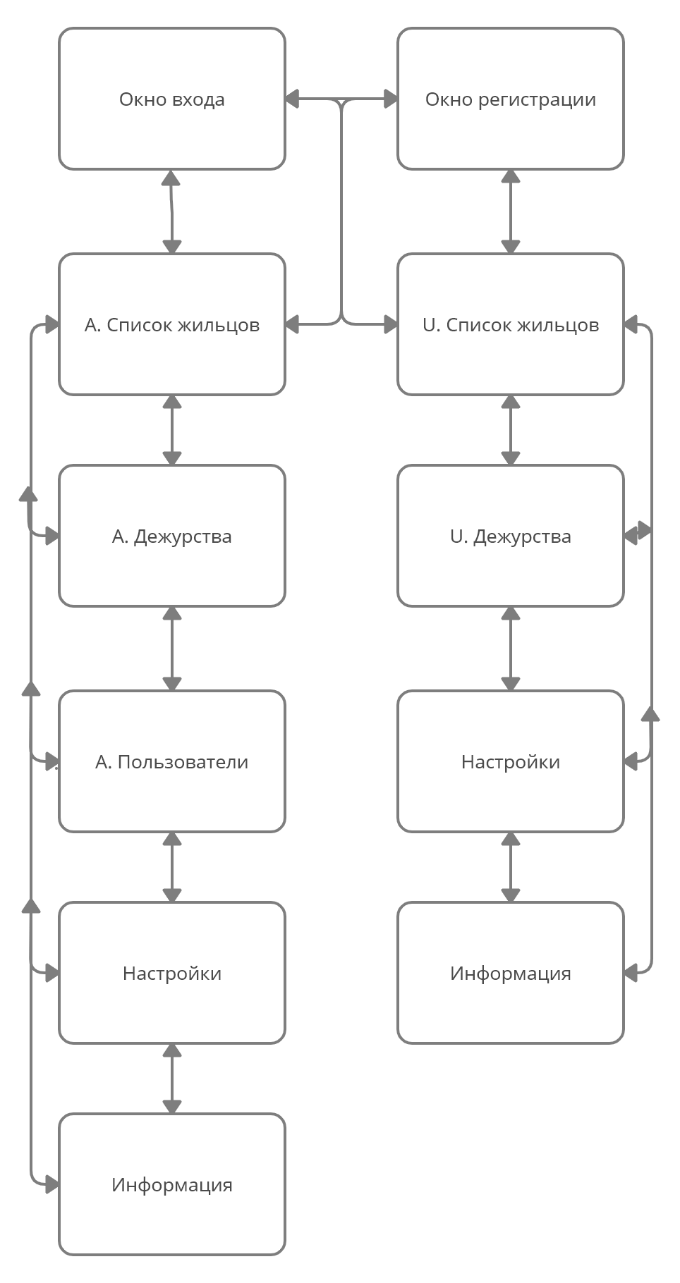


Рис. 2.1 – Диаграмма окон

В окне входа пользователь вводит данные и получает доступ к окну со списком жильцов (в зависимости от пользователя будет доступен разный функционал, на диаграмме разные пользователи помечены аббревиатурами A и U для администратора и пользователя соответственно).

Окно со списком жильцов для администратора позволяет редактировать, удалять, а также добавлять новых жильцов. Также присутствует поиск по жильцам и сортировка по выбранному столбцу.

Окно с дежурствами для администратора дает возможность записать любого человека на дежурство на любое доступное время.

Окно со списком жильцов для обычного пользователя предполагает возможность просмотра всех жильцов общежития, выполнить поиск и сортировки по ним.

Окно с дежурствами для пользователя дает возможность записать себя на любое доступное время на дежурство.

Окно с пользователями для администратора позволяет просмотреть всех зарегистрированных пользователей, а также выполнить поиск и сортировку по ним.

Окно настроек предполагает возможность изменения логина, либо пароля для пользователя.

Окно информации позволяет посмотреть более подробную информацию об общежитии.

## 2.4 Ресурсы проекта

К ресурсам проекта относятся иконки, окна и пользовательские элементы управления.

Ресурсы проекта:

* 1.jpg – изображение общежития;
* 2.jpg – изображение общежития;
* 3.jpg – изображение общежития;
* 4.jpg – изображение общежития;
* 5.jpg – изображение общежития;
* 6.jpg – изображение общежития;
* duty.png – иконка для пункта меню «Дежурства»;
* human.png – иконка для пункта меню «Список жильцов»;
* magnifier.png – иконка для пункта меню «Информация»;
* setting.png – иконка для пункта меню «Настройки»;
* users.png – иконка для пункта меню «Список пользователей»
* icon.png – иконка приложения;
* StartWindow.xaml и StartWindow.xaml.cs – стартовое окно;
* MainWindow.xaml и MainWindow.xaml.cs – главное окно администратора;
* UserMainWIndow.xaml и UserMainWindow.xaml.cs – главное окно пользователя;
* DutiesListFrame.xaml и DutiesListFrame.xaml .cs и – администраторский список дежурств;
* TenantsDataGridFrame.xaml и TenantsDataGridFrame.xaml.cs – администраторский список жильцов;
* UserDataGridFrame.xaml и UserDataGridFrame.xaml.cs –список пользователей;
* UserRegOnDuty.xaml и UserRegOnDuty.xaml.cs – пользовательская запись на дежурства;
* UserTenantsDataGridFtame.xaml и UserTenantsDataGridFtame.xaml.cs – пользовательский список жильцов;
* DormitoryInfoFrame.xaml и DormitoryInfoFrame.xaml.cs – информация об общежитии;
* EditAccountFrame.xaml и EditAccountFrame.xaml.cs – редактирование данных аккаунта;
* AuthFrame.xaml и AuthFrame.xaml.cs – вход;
* RegisterFrame.xaml и RegisterFrame.xaml.cs – регистрация.

## 2.5 Пример варианта использования

Диаграмма использования «Пользователь» и «Администратор» предоставлены в приложении Б

Для пользователя предусмотрен следующий сценарий.

На окне авторизации пользователь должен ввести свои данные или же открыть окно регистрации. При успешной регистрации пользователя он снова попадет на окно авторизации. Если зарегистрированный пользователь ввёл верные данные он переходит на главное окно с открытой вкладкой со списком жильцов общежития. На ней он может просмотреть нужную ему информацию. Также в бургер-меню слева он может выбрать любую другую вкладку, на которых он может выполнить ряд различных действий. Перечень сценариев содержит в себе множество вариантов событий, среди которых находятся такие как изменение своего аккаунта, запись на дежурство, просмотр информации об общежитии, поиск различной информации.

Для администратора предусмотрен следующий сценарий.

На окне авторизации пользователь должен ввести свои данные. Если данные введены корректно, администратор попадает на окно со списком жильцом общежития администратора, в противном случае он пытается ввести данные снова или же покидает приложение. Далее он может выполнить ряд действий по изменению имеющихся данных, а также по поиску оных. Слева располагается бургер-меню, которое аналогично пользовательскому, однако имеет один дополнительный пункт со списком пользователей и ряд дополнительных функций, которые обычному пользователю недоступны. Некоторые сценарии поведения администратора включают в себя редактирование списка жильцов, либо дежурств, просмотр пользователей, зарегистрированных в приложении, изменение данных своего аккаунта и прочее.

Диаграмма последовательностей для авторизации пользователя отображена в приложении В.

## 2.6 Структура проекта

Структура проекта – «скелет» любой программы. Он состоит из различных элементов. Это могут быть ресурсы, классы, интерфейсы, формы и прочее.

Класс Tenant содержит в себе базовые свойства жильца общежития.

Класс User содержит в себе базовые свойства пользователя программы.

Класс Duty содержит в себе базовые свойства для представления дежурства.

Класс ApplicationContext ложится в основу связи программы с базой данных.

Класс DataWorker содержит в себе методы для работы с базой данных.

Класс Validator является ключевым для валидации введенных данных и проверяет их на корректность.

Класс ValidatingException является реализацией пользовательской ошибки для корректного функционирования класса Validator.

Классы с расширением «.xaml» и «.xaml.cs» являются классами представления. Они содержат в себе всю логику, необходимую для корректного отображения пользовательского интерфейса.

Структура проекта представлена диаграммой классов в графических материалах.

3. РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ МОДЕЛИ И МОДЕЛИ ПС

## 3.1 Взаимодействие с базой данных

Чтобы осуществлять связь между базой данных и приложением на C# необходим посредник. И именно таким посредником является технология Entity Framework.

Entity Framework — это решение для работы с базами данных, которое используется в программировании на языках семейства.NET. Оно позволяет взаимодействовать с СУБД с помощью сущностей (entity), а не таблиц.

Для работы с базой данных были созданы два класса: ApplicationContext и DataWorker. Класс ApplicationContext является связью между самой реляционной базой данных (MS SQL Server) и сущностями (списками объектов). Также он настраивает базу данных и создает ее, если не существует. В свою очередь статический класс DataWorker позволяет удобно взаимодействовать с базой данных, просто передавая в функции необходимые объекты. Программный код класса ApplicationContext представлен на рис 3.1.

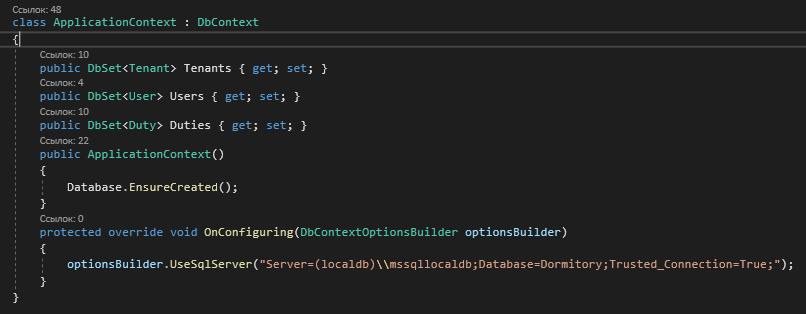


Рис. 3.1 – Класс ApplicationContext

## 3.2 DataGrid

DataGrid является ключевым компонентом приложения, так как он заточен под работу с базой данных и в удобном формате позволяет выводить, редактировать и сортировать различную информацию.

В приложении присутствует 5 элементов DataGrid, которые настроены абсолютно разным образом. Одной из ключевых особенностей этого элемента управления является привязка данных к некоторому списку. Пример из проекта представлен в приложении Г.

## 3.3 Валидация

Валидация совершается практически для любых данных, введенных пользователем. Было удобно создать класс Validator, который отвечал бы за эту функцию. Также для корректной и удобной работы класса Validator было создано пользовательское исключение ValidatingException (рис. 3.2), которое соответствует функционалу обычного исключения, однако имеет одно отличие: в нем хранится еще и тип ошибки (ошибка пароля или ошибка логина).



Рис. 3.2 – Класс ValidatingException

После вылавливания ошибки типа ValidatingException класс Validator передает ее по иерархии выше, и она обрабатывается на уровне окна. При неверном вводе информации поле и иконка рядом с полем подсвечивается красным, также появляется подсказка с текстом ошибки.

Программный код класса Validator представлен в приложении Д.

## 3.4 Схема и описание авторизации

Авторизация в приложении происходит следующим образом.

После открытия окна пользователь должен ввести свой логин и пароль. После нажатия на кнопку «Войти» данные, введенные с клавиатуры, передаются в класс DataWorker для того, чтобы попробовать найти в базе данных подобного пользователя. Это происходит при помощи каждой Linq запроса, что увеличивает быстродействие программы, однако первый вход в базу данных обеспечивает небольшое подтормаживание.

Блок-схема данного процесса представлена в приложении Е.

## 3.5 Использованные готовые решения

В проекте используется два сторонних решения для бургер-меню и дизайна: HamburgerMenu и MaterialDesign.

HamburgerMenu – простое настраиваемое средство для создания бургер-меню с его помощью происходит перемещение по страницам проекта.

MaterialDesign - стиль графического дизайна интерфейсов программного обеспечения и приложений, разработанный компанией Google. Впервые представлен на конференции Google I/O 25 июня 2014 года. Стиль расширяет идею «карточек», появившуюся в Google Now, более широким применением строгих макетов, анимаций и переходов, отступов и эффектов глубины (света и тени). По идее графических дизайнеров Google, у приложений не должно быть острых углов, карточки должны переключаться между собой плавно и практически незаметно.

## 3.6 Перемещение по страницам

Перемещение по страницам приложения происходит за счет наличия всего одного окна и бургер-меню в нем. В окне существует пустая область, которая заполняется динамически при помощи пользовательских элементов управления, которые и являются страницами. Изначально они все размещаются в окне, однако все скрыты, кроме одной и при нажатии на пункт меню открывается другое окно, а предыдущее скрывается.

4. ТЕСТИРОВАНИЕ

## 4.1 Тестирование валидации

Валидация – это проверка продукта, процесса или системы на соответствие требованиям клиента.

В нашем случае валидация – это проверка информации, вводимой пользователем, на соответствие программным требованиям.

В окне авторизации примером валидации является ввод логина и пароля. Недопустимость ввода некорректных данных показана на рисунке 4.1.

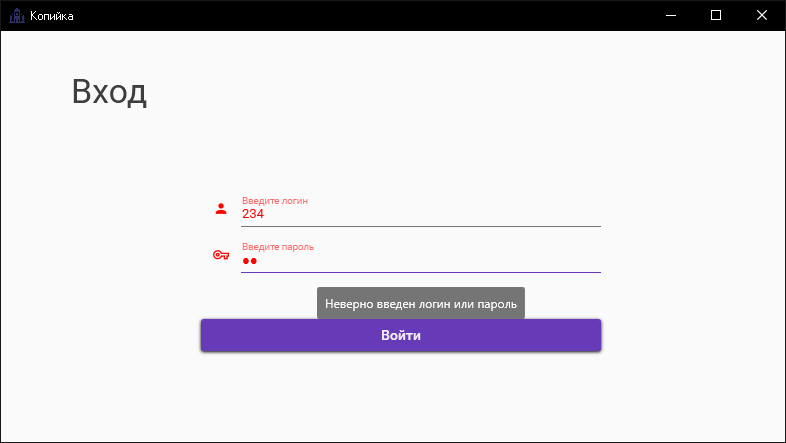


Рис 4.1 – Валидация логина и пароля

## 4.2 Тестирование исключительных ситуаций

Исключительная ситуация (exception) – это любая ошибка или ошибочное условие, возникающее в процессе выполнения программы. Когда программа обнаруживает ошибку, она генерирует исключительную ситуацию.

Исключительные ситуации в данном приложении могли бы возникнуть при попытке создания жильца от лица администратора с недостающими данными (рис. 4.2). Однако это не вызовет ошибку приложения будет ждать пока администратор добавит недостающие данные в будущем.

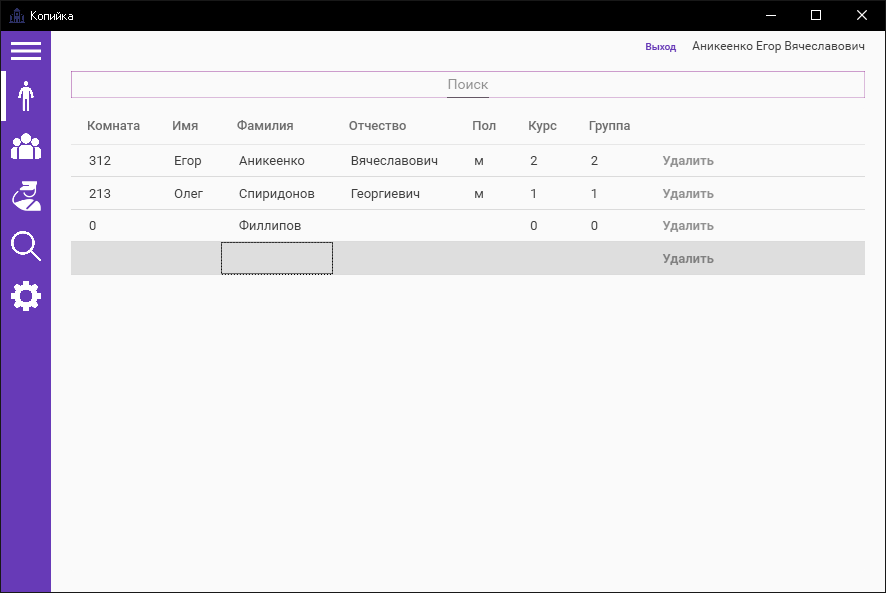


Рис. 4.2 – Добавление жильца с недостающими данными

## 4.3 Использование приложения обычным пользователем

При запуске приложения пользователь видит окно авторизации приложения, представленное на рисунке 4.3.

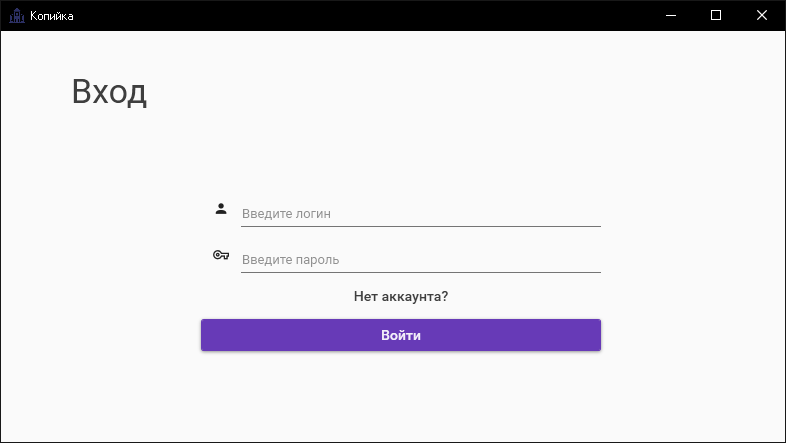


Рис. 4.3 – Окно входа

При нажатии на кнопку «Нет аккаунта?» пользователь сможет зарегистрироваться в программном средстве и начать пользоваться приложением, однако обязательным условием регистрации является проживание в общежитии, поэтому пользователь должен выбрать свое ФИО в списке при регистрации без последующей возможности изменения оного. Окно регистрации представлено на рисунке 4.4.

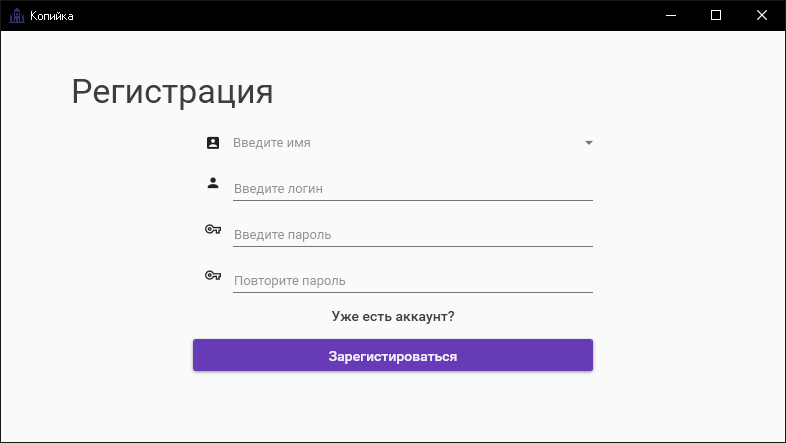


Рис. 4.4 – Окно регистрации

После авторизации пользователь видит список жильцов общежития (рис. 4.5).

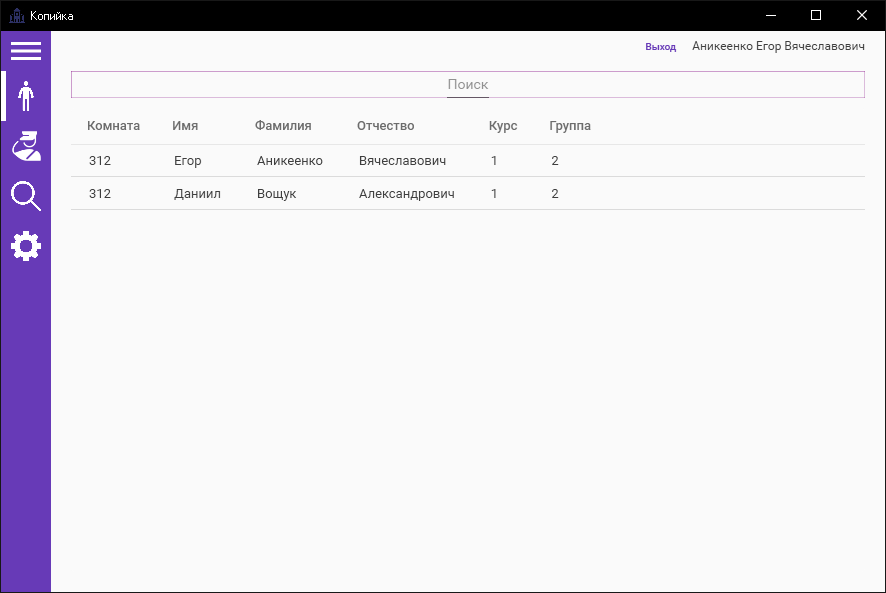


Рис. 4.5 – Окно со списком жильцов общежития

Далее у пользователя имеется ряд операций, которые пользователь может выполнить, некоторые из них представлены на рис. 4.6.

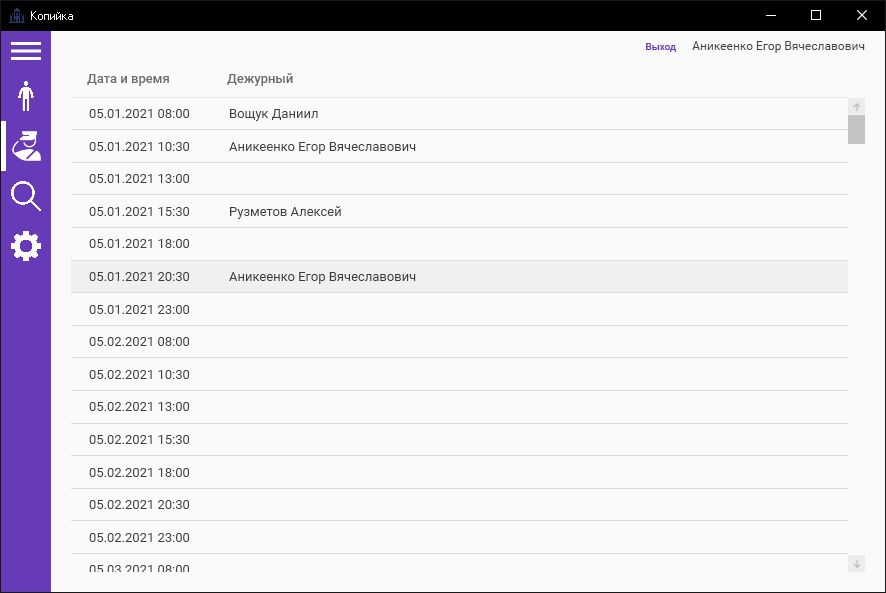
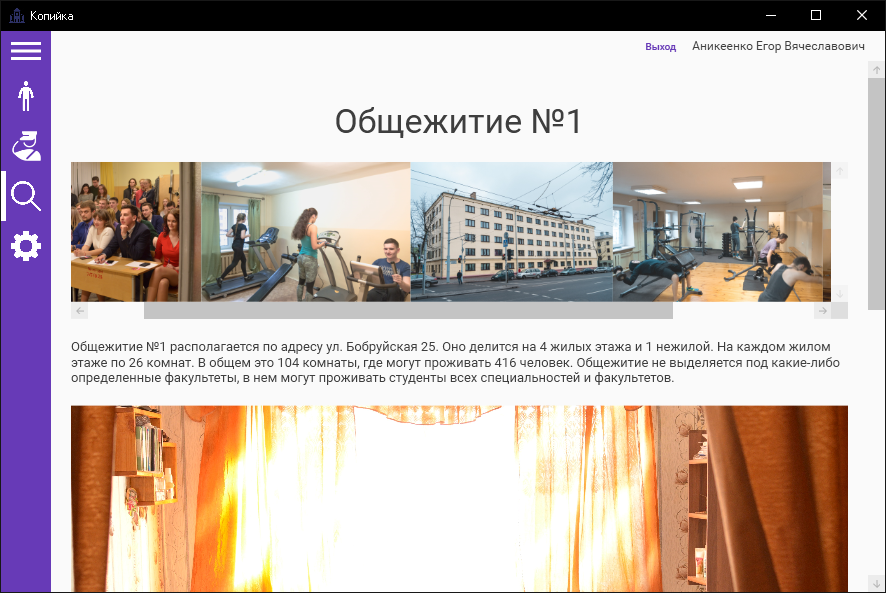


Рис. 4.6 – Примеры возможных действий

## 4.4 Использование приложения администратором

При запуске приложения пользователь видит окно авторизации приложения, представленное на рисунке 4.7.

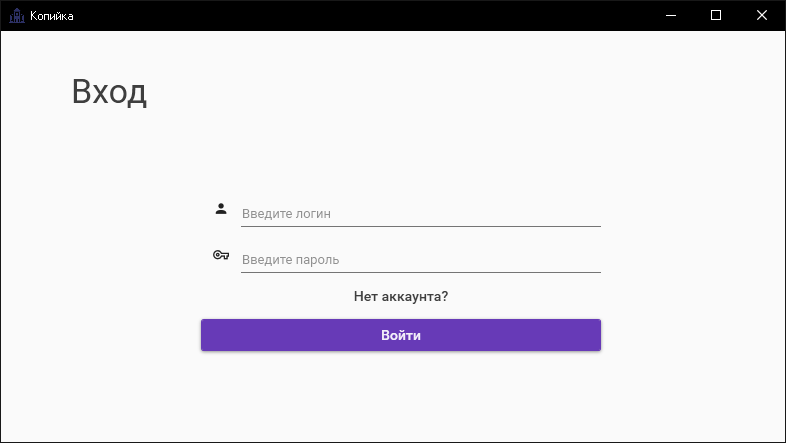


Рис. 4.7 – Окно входа

При нажатии на кнопку «Нет аккаунта?» пользователь сможет зарегистрироваться в программном средстве и начать пользоваться приложением, однако обязательным условием регистрации является проживание в общежитии, поэтому пользователь должен выбрать свое ФИО в списке при регистрации без последующей возможности изменения оного. Окно регистрации представлено на рисунке 4.8.

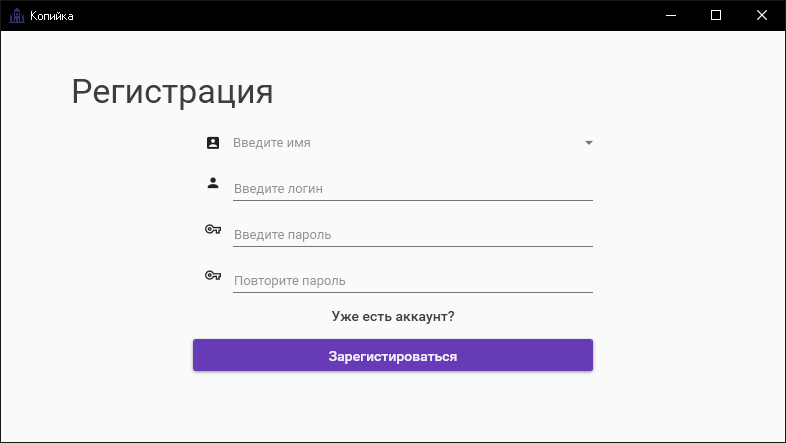


Рис. 4.8 – Окно регистрации

После авторизации пользователь видит список жильцов общежития.

Далее администратор может производить те же действия, что и пользователь, однако имеет больше возможностей, одно из которых просмотр пользователей приложения без возможности редактирования (рис. 4.9).

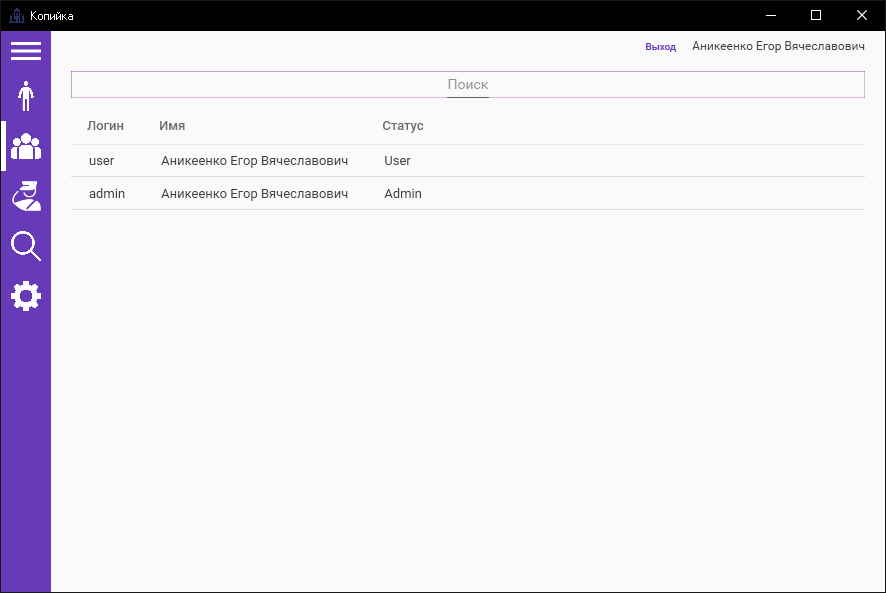


Рис. 4.9 – Окно просмотра пользователей

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта были рассмотрены основные аналоги программного средства, их преимущества и недостатки, были просмотрены и изучены их возможности. Произведен анализ и обзор теоретического материала по данной тематике курсового проекта.

Также были изучены основные технологии и сторонние библиотеки, которые применялись в разработке данного программного средства. В качестве системы управления базой данных был выбран и использован Microsoft SQL Server 2012.

Данный проект был разработан для использования его пользователями, чтобы они смогли администрировать общежитие или же в удобном формате читать актуальную информацию, которая им необходима. Главной задачей было сделать простое и удобное приложение. Для этого было разработано 9 окон: «Регистрация», «Авторизация», «Список жильцов», «Список пользователей», «Информация», «Настройки», «Дежурства», а также «Список жильцов» и «Дежурства» для администратора, которые предоставляют больше возможностей.

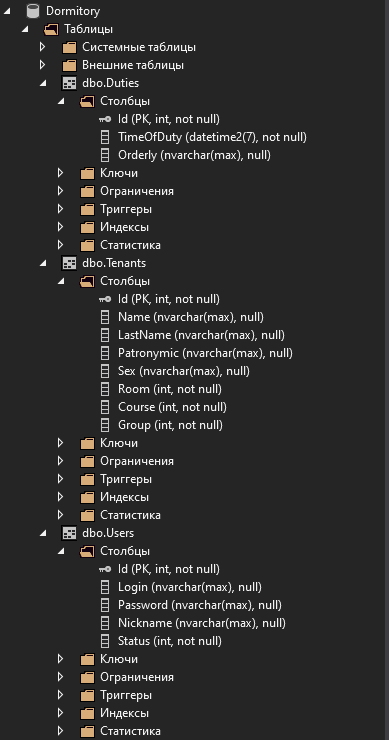
Так как данный опыт разработки подобных приложений на технологии WPF был первым, естественно, есть приложения в разы лучше. Но, тем не менее, это в очередной раз свидетельствует о том, что есть к чему стремиться.

Список используемых источников

1. Информационный сайт: контроль питомца [Электронный ресурс] kusochek.by – https://kusochek.by/biblioteka/teoriya-praktika-kormleniya-zhivotnih/ – Дата доступа: 24.04.2021.
2. Информационный ресурс: Dog Log Book – Режим доступа: play.google.com/store/apps/details?id=com.doglogbook – Дата доступа: 24.04.2021.
3. Информационный ресурс: DogLog – Режим доступа: play.google.com/store/apps/details?id=com.doglogbook&hl=ru/. – Дата доступа: 24.04.2021.
4. Информационный ресурс: Дневник питомца – Режим доступа: play.google.com/store/apps/details?id=com.petsdiary&hl=ru/. – Дата доступа: 24.04.2021.
5. Сайт о программировании: MVVM [Электронный ресурс] / medium.com – Режим доступа: https://medium.com/@losiennik/mvvm-architectural-pattern-on-ios-using-swift-a7b039e3101 – Дата доступа: 24.04.2021.
6. Информационный сайт: базы данных [Электронный ресурс] kusochek.by – inform.sch117nn.edusite.ru/DswMedia/klyaks-\_netuchitelyutestyibazyidannyix.htm – Дата доступа: 24.05.2020.
7. Сайт о программировании Metanit: Введение в ADO.NET [Электронный ресурс]– Режим доступа: https://metanit.com/sharp/adonet/1.1.php – Дата доступа: 24.04.2021.
8. Информационный сайт: Валидация [Электронный ресурс] myrouble.ru – https://myrouble.ru/validation/ – Дата доступа: 24.04.2021.

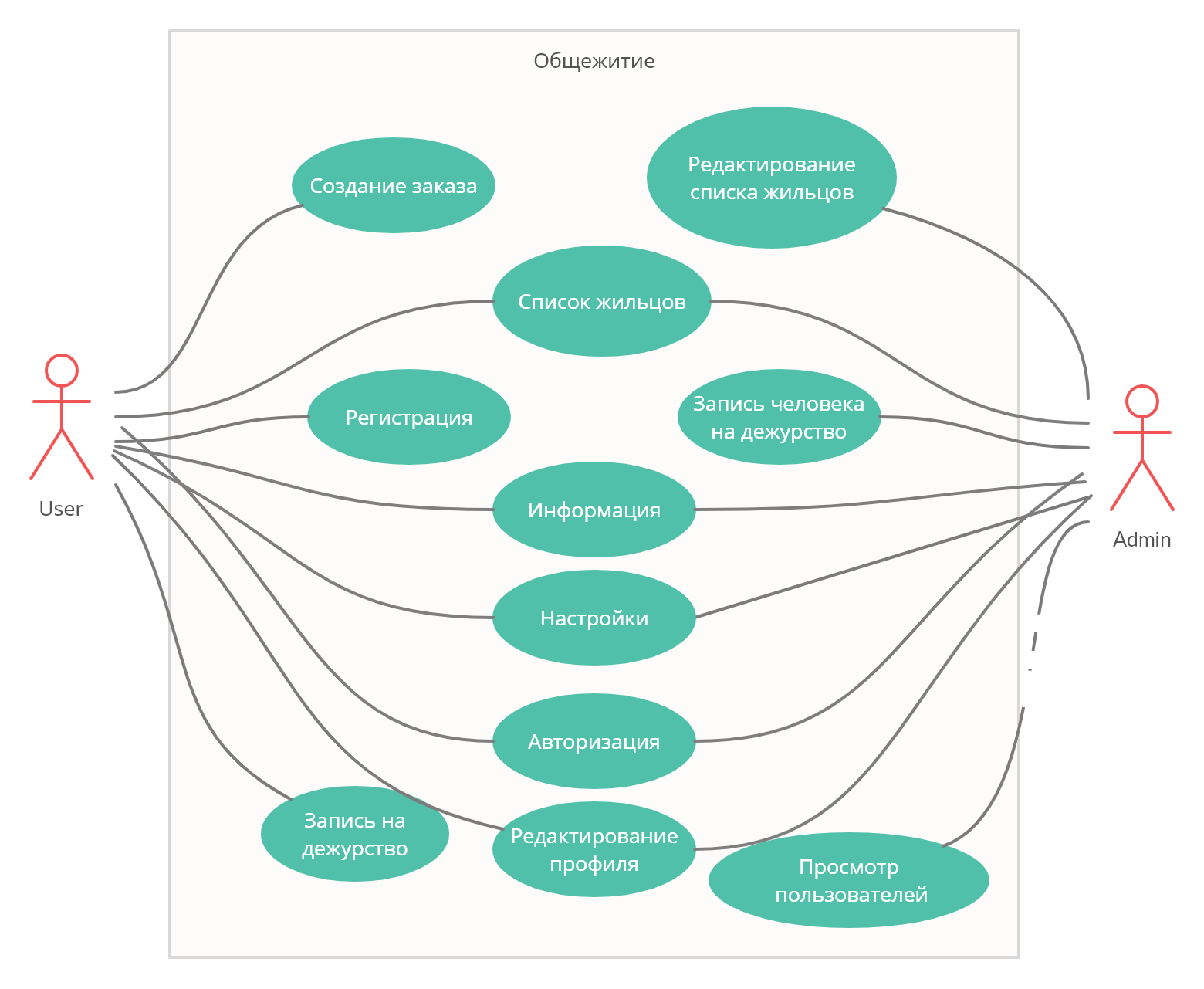
ПРИЛОЖЕНИЕ А

Логическая структура базы данных



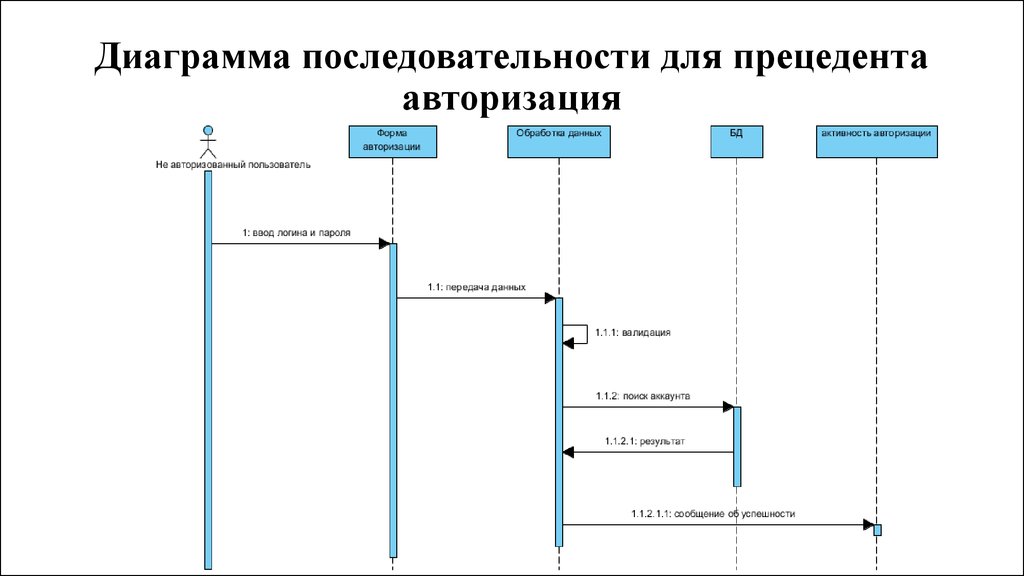
ПРИЛОЖЕНИЕ Б

Диаграмма использования для пользователя и администратора



ПРИЛОЖЕНИЕ В

Диаграмма последовательности авторизации



ПРИЛОЖЕНИЕ Г

Листинг 1. Пример использования DataGrid на примере списка жильцов общежития

<DataGrid Grid.Row="1" x:Name="TenantsDataGrid" AutoGenerateColumns="False" Margin="20 0 20 20" ItemsSource="{Binding TenantsList}">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Path=Room}">

<DataGridTextColumn.Header>

<TextBlock Text="Комната"/>

</DataGridTextColumn.Header>

</DataGridTextColumn>

<DataGridTextColumn Binding="{Binding Path=Name}">

<DataGridTextColumn.Header>

<TextBlock Text="Имя" />

</DataGridTextColumn.Header>

</DataGridTextColumn>

<DataGridTextColumn Binding="{Binding Path=LastName}">

<DataGridTextColumn.Header>

<TextBlock Text="Фамилия" />

</DataGridTextColumn.Header>

</DataGridTextColumn>

<DataGridTextColumn Binding="{Binding Path=Patronymic}">

<DataGridTextColumn.Header>

<TextBlock Text="Отчество" />

</DataGridTextColumn.Header>

</DataGridTextColumn>

<DataGridTextColumn Binding="{Binding Path=Sex}">

<DataGridTextColumn.Header>

<TextBlock Text="Пол" />

</DataGridTextColumn.Header>

</DataGridTextColumn>

<DataGridTextColumn Binding="{Binding Path=Course}">

<DataGridTextColumn.Header>

<TextBlock Text="Курс" />

</DataGridTextColumn.Header>

</DataGridTextColumn>

<DataGridTextColumn Binding="{Binding Path=Group}">

<DataGridTextColumn.Header>

<TextBlock Text="Группа" />

</DataGridTextColumn.Header>

</DataGridTextColumn>

<DataGridTemplateColumn IsReadOnly="True">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Cursor="Hand" Content="Удалить" Command="Delete" Style="{StaticResource {x:Static ToolBar.ButtonStyleKey}}" BorderThickness="0" Margin="0" Padding="0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns></DataGrid>

ПРИЛОЖЕНИЕ Д

Листинг 2. Класс для валидации логина и пароля

public static class Validator

{

public static bool ValidateLogin(string str)

{

string symbols = "!@#$%^^&\*()\_+=-?.,<>:;\\\"'`~ ";

if (str.Length < 4)

throw new ValidatingException("Длина строки должна превышать 4 символа", ValidatingErrorTypes.StrError);

foreach(var ch in str)

{

foreach(var symbol in symbols)

{

if (ch == symbol)

throw new ValidatingException("Введены запрещенные символы", ValidatingErrorTypes.StrError);

}

}

string HashedPassword = Convert.ToBase64String(MD5.Create().ComputeHash(Encoding.UTF8.GetBytes(str)));

using(ApplicationContext db = new ApplicationContext())

{

var logins = DataWorker.GetAllUsers().Where(x => x.Login == str) as List<User>;

if (logins != null)

throw new ValidatingException("Такой пользователь уже существует", ValidatingErrorTypes.StrError);

}

return true;

}

public static bool ValidatePassword(string password)

{

if (password.Length < 0)

throw new ValidatingException("Длина пароля должна быть больше 8 символов", ValidatingErrorTypes.PassError);

else if (password.Length > 16)

throw new ValidatingException("Длина пароля не должна првевышать 16 символов", ValidatingErrorTypes.PassError);

return true;

}

}

ПРИЛОЖЕНИЕ Е

Блок-схема для авторизации

