Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский государственный технический университет**

**им. Н.Э. Баумана»**

Национальный исследовательский университет техники и технологий

**(МГТУ им. Н.Э.Баумана)**

*Факультет* ***«Робототехника и комплексная автоматизация» (РК)***

*Кафедра* ***«Системы автоматизированного проектирования» (РК-6)***

Домашнее задание №1

по курсу «Разработка информационной системы «Авиабилеты»

Студент группы РК6-52:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Мигранов Р.Р.)

Руководитель работы:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Пивоварова Н.В.)

Москва 2017

Описание предметной области.

Автоматизация бизнеса — одно из главных условий успеха любой организации, независимо от ее структуры, масштабов и сферы деятельности. Её результатом является высвобождение человеческих и финансовых ресурсов для повышения производительности труда и эффективности стратегического управления.

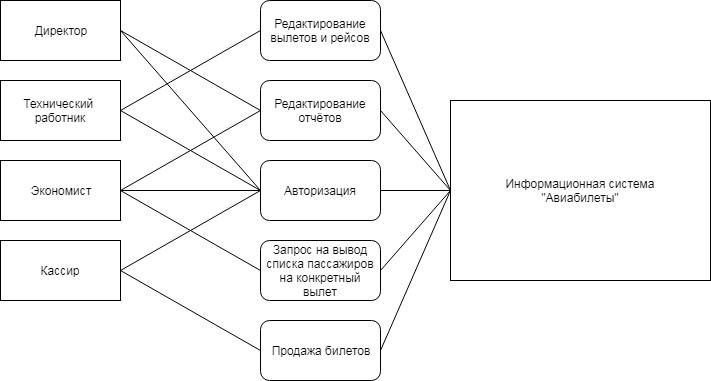
В нашем случае авиакомпании необходима удобная и функциональная информационная система, способная выполнять ряд задач по автоматизации конкретных бизнес – процессов.

Для каждой группы сотрудников разработаны свои доступные варианты использования информационной системы.

1. Конечные пользователи информационной системы:

* директор авиакомпании;
* технический работник;
* экономист;
* кассир.

1. Варианты использования информационной системы конечными пользователями:



Все варианты использования информационной системы подразумевают обращение к базе данных.

1. Основным вариантом использования информационной системы (основной бизнес-процесс) является продажа билетов кассирами. Данный вариант имеет сложную структуру, которая включает в себя несколько обращений к базе данных и динамическое обновление страницы. Также он реализует основную идею, ради которой разрабатывается система, - увеличение прибыли компании за счет расширения функционала информационной системы.
2. Главные успешные сценарии вариантов использования и расширения к ним.
3. *Редактирование вылетов и рейсов.*
4. Технический работник запрашивает страницу редактирования.
5. Система присылает форму с выбором редактирования вылетов или рейсов.
6. Пользователь выбирает нужную категорию, отправляет это в систему.
7. Система передаёт пользователю список вылетов/рейсов.
8. Пользователь выполняет запрос добавления/изменения поля.
9. Пользователь получает от системы форму добавления/изменения.
10. Пользователь отправляет данные для редактирования системе.
11. Сообщение об успешном выполнении пользователю с возможностью выхода в главное меню.

Расширение:

1. а) Данные в системе уже существуют.

Система говорит об ошибке и предлагает:

1. вернуться в главное меню;
2. повторить пункт II.

б) Неверные данные в форме редактирования.

Система говорит об ошибке и предлагает:

1. вернуться в главное меню;
2. повторить пункт II.
3. *Редактирование отчетов по доходам от проданных билетов.*
4. Запуск процесса редактирования отчётов.
5. Система присылает пользователю список, в котором будут указаны месяц и год созданных отчётов. Напротив каждого находится кнопка “показать” и “удалить”. Вверху страницы кнопка “Добавить отчёт”. Пользователь выбирает её.
6. Система в ответ присылает форму с вводом месяца и года.
7. Пользователь отправляет форму с введённым месяцем и годом системе.
8. Система сообщает пользователю, что отчёт создан и выводит полученную таблицу отчёта пользователю с возможностью выхода в главное меню.

Расширение:

1. а) Система сообщает пользователю, что билеты в данный месяц и год не продавались и предлагает:
2. вернуться в главное меню;
3. повторить пункт II.

б) Система сообщает пользователю, что отчёт уже создан, выводит его на экран и предлагает вернуться в главное меню.

1. *Авторизация.*
   * 1. Пользователь запускает процесс авторизации.
     2. Система в ответ на это посылает HTML – страницу с формой ввода логина и пароля.
     3. Пользователь вводит логин и пароль и отправляет это системе.
     4. Пользователь входит в систему.

Расширение:

1. а) СУБД отвечает системе, что пользователь не найден.

Система сообщает пользователю о том, что логин неверный.

1. Возврат к пунку II.
2. Возврат в главное меню.

б) СУБД отвечает системе, что пароль неверный.

Система сообщает пользователю о том, что пароль неверный.

1. Возврат к пунку II.
2. Возврат в главное меню.
3. *Продажа билетов.*
4. Кассир делает системе запрос на продажу билетов.
5. Система присылает HTML – страницу с формой ввода аэропорта вылета, аэропорта прилёта, даты вылета.
6. Кассир заполняет данные поля и отправляет это в систему.
7. Система присылает список всех вылеты на указанную дату.
8. Кассир запрашивает страницу с продажей билетов на необходимый вылет.
9. Система присылает HTML – страницу, где указаны: аэропорт вылета, аэропорт прилёта, дата и время вылета, дата и время прилёта, стоимость 1 билета и форма для ввода количества билетов.
10. Система отвечает пользователю, что билет(ы) продан(ы) успешно.

Расширение:

1. а) Вылетов на данную дату нет. Система сообщает об этом пользователю и предлагает:

a) вернуться к п. 2 с целью ввести другую дату вылета;

b) вернуться в главное меню.

б) Подходящего клиенту вылета нет. Система предлагает:

1. вернуться к п. 2 с целью ввести другую дату вылета;
2. вернуться в главное меню.
3. *Показать список пассажиров на конкретный вылет.*
4. Пользователь делает запрос системе на получение списка пассажиров на конкретный вылет.
5. Система в ответ присылает HTML страницу с полем ввода даты вылета.
6. Пользователь вводит дату вылета и отправляет это системе.
7. Система отправляет пользователю список вылетов.
8. Пользователь выбирает нужный вылет и отправляет его системе.
9. Система возвращает пользователю список пассажиров.

Расширение:

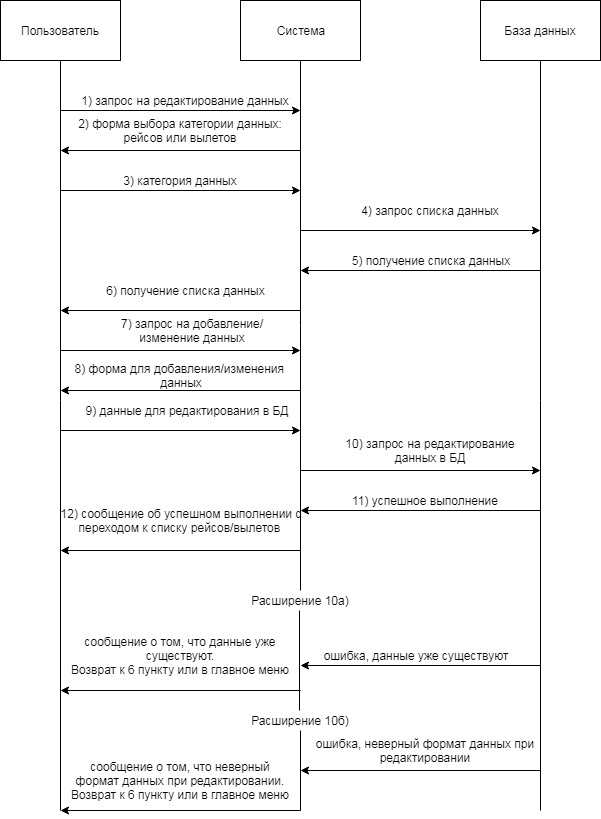
1. а) Вылетов на указанную дату нет.

Система предлагает:

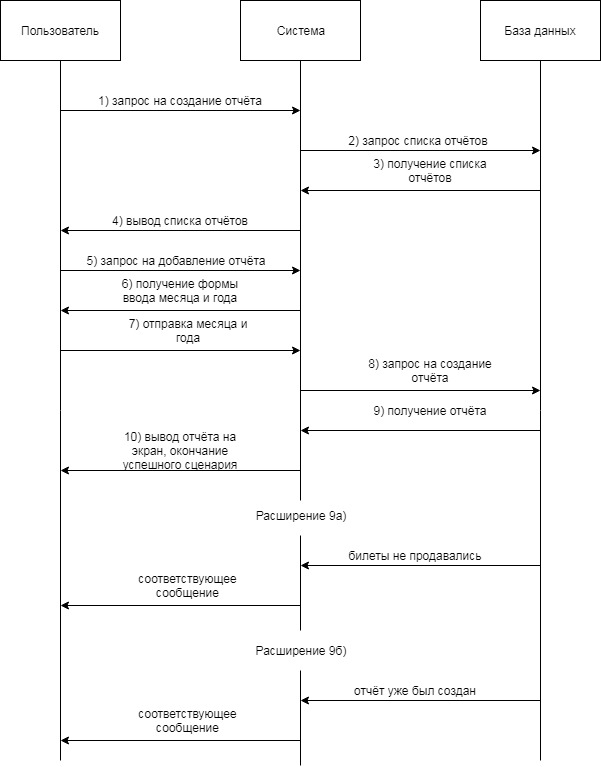
1. вернуться к п. 2 и ввести другую дату;
2. вернуться в главное меню.
3. а) Список пассажиров пуст. Система сообщает об этом пользователю и предлагает вернуться в главное меню.

Системные UML-диаграммы последовательности пяти сценариев (контроллеров).

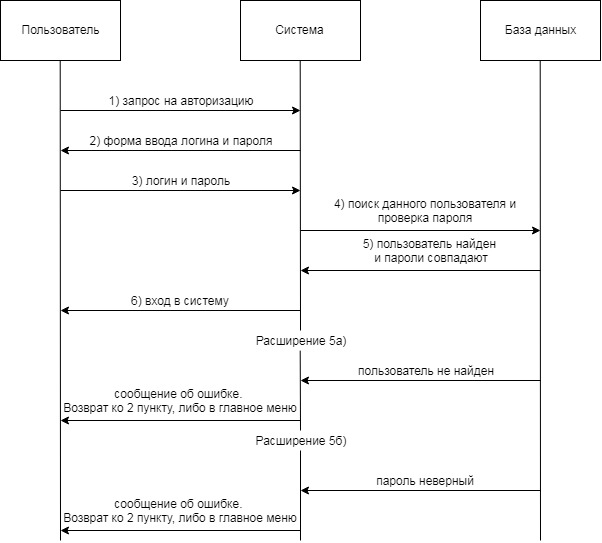
1. *Редактирование вылетов и рейсов.*



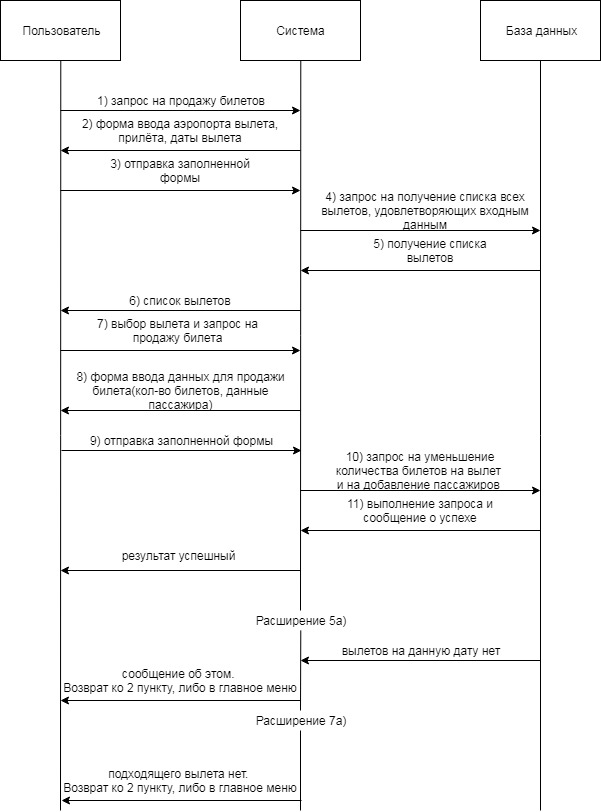
1. *Отчеты по доходам от проданных билетов.*



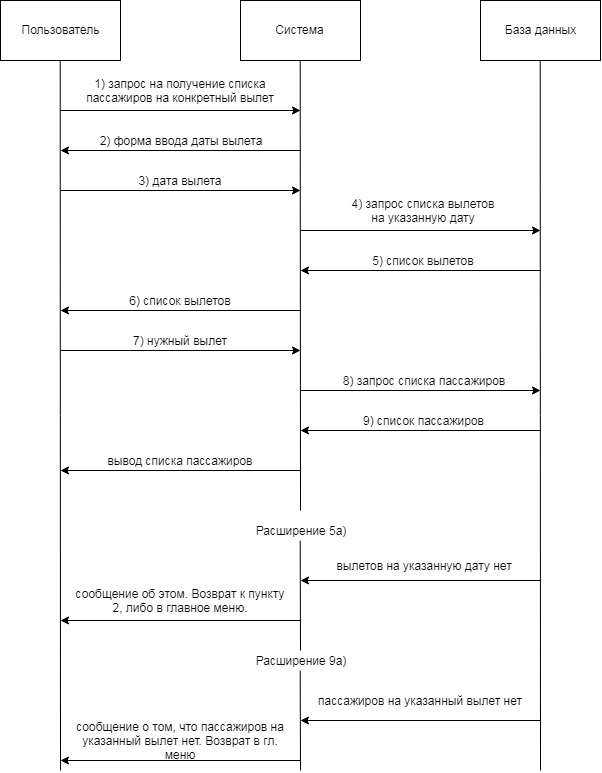
1. *Авторизация.*

**

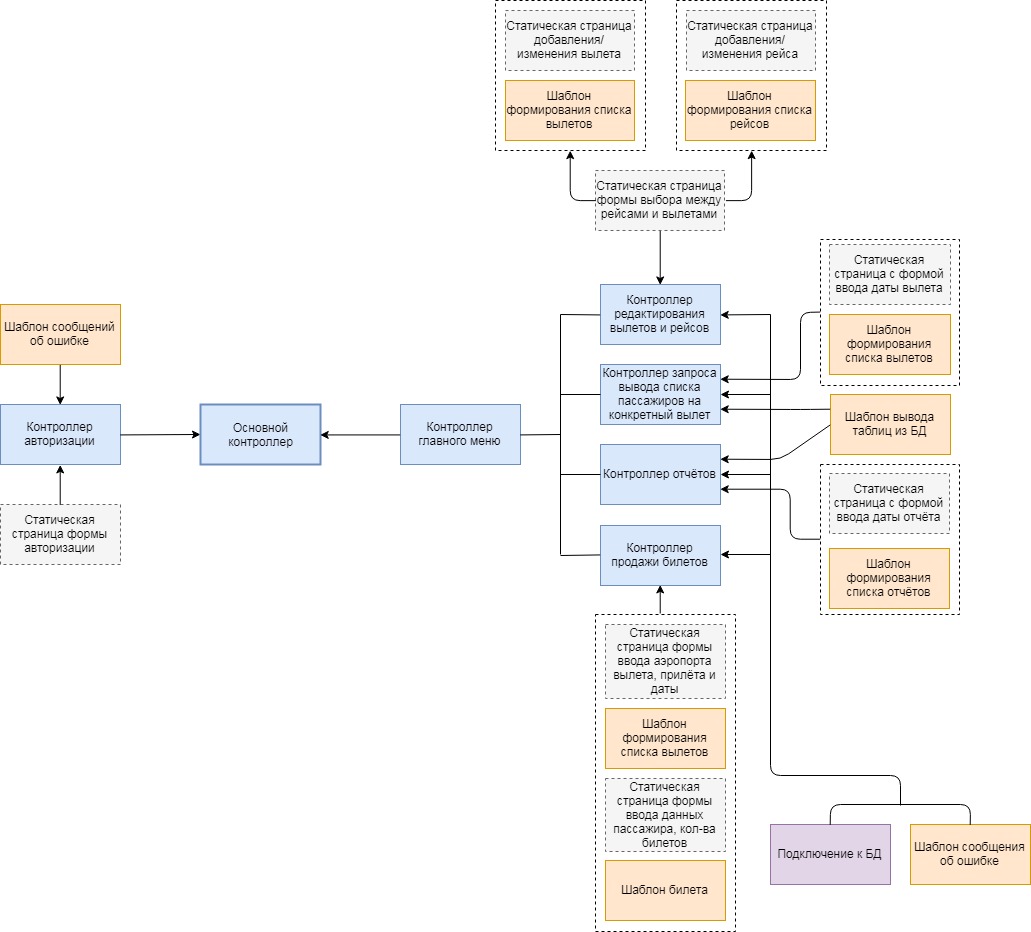
1. *Продажа билетов.*



1. *Показать список пассажиров на конкретный вылет.*



Набор шаблонов и контроллеров.



Контроллер авторизации

Формирует страницу, содержащую форму авторизации (ввод логина и пароля).

После получения заполненной формы, проверяет наличие логина в базе данных. Если такой логин зарегистрирован, то сравнивает пароли. Иначе, отправляет сообщение об ошибке некорректного ввода данных и снова выдает форму авторизации. Если логин не зарегистрирован, выдает сообщение об ошибке некорректного ввода данных и снова выдает форму авторизации. При успешном совпадении вводимых логина/пароля, передает управление основному контроллеру.

Основной контроллер

Определяет, авторизирован ли пользователь:

Да) Передает управление контроллеру главного меню;

Нет) Передает управление контроллеру авторизации.

Контроллер главного меню

В зависимости от роли определяет, к каким пунктам меню пользователь имеет доступ:

Директор: Отчеты.

Технический работник: Редактирование вылетов и рейсов.

Экономист: Отчёты, Запросы.

Кассир: Продажа билетов.

Формирует страницу, содержащую доступные пункты меню.

При выборе одного из пунктов, управление передается соответствующему контроллеру.

Контроллер отчетов

Формирует страницу содержащую список доступных отчетов.

При выборе пользователем раздела отчета, контроллер формирует форму для отчета (Если формы не предусмотрено, то пропускает этот пункт). После получения заполненной формы БД проверяет корректность введенных данных. В случае неверного ввода возвращает сообщение об ошибке и отправляет форму заново. При верном заполнении формы выполняет запрос и выводит данные в виде таблицы. Если в БД нет данных по форме, выводит сообщение об отсутствии запрашиваемых данных и возвращает форму отчета.

Контроллер запросов

Формирует страницу содержащую список доступных запросов.

При выборе пользователем нужного запроса, контроллер формирует форму для конкретного запроса (Если формы не предусмотрено, то пропускает этот пункт). После получения заполненной формы проверяет корректность введенных данных. В случае неверного ввода возвращает сообщение об ошибке и отправляет форму заново. При верном заполнении формы выполняет запрос и выводит данные в виде таблицы. Если в БД нет данных по форме, выводит сообщение об отсутствии запрашиваемых данных и возвращает форму запроса.

Контроллер редактирования расписания вылетов и рейсов

Контроллер создаёт форму, в которой пользователь может выбрать дату и категорию – вылеты или рейсы, по которой будет выведено расписание. После получения даты система формирует запрос и выводит полученное расписание, в котором пользователь может выбрать поле для редактирования или добавить новое. После выбора поля (или добавления) пользователю отправляется форма для редактирования. Если пользователь неверно заполнит форму, то будет выведено сообщение об ошибке и возвращена форма. При верном заполнении данные будут занесены/переписаны в БД. При любых ошибках с БД контроллер также выводит сообщение об ошибках.

Контроллер продажи билетов

Контроллер создаёт форму для выбора даты (по умолчанию текущая). После получения даты формирует запрос и выводит расписание. Когда от пользователя будут получены данные по аэропорту и времени, контроллер сформирует запрос и выведет форму для ввода данных пассажира. После получения данных контроллер заносит данные о купленном билете в БД.

Система авторизации.

Система авторизации делится на два уровня:

* авторизация в информационной системе;
* авторизация в базе данных.

Для авторизации в информационной системе, в базе данных предварительно создается таблица Authorization, в которой хранятся следующие поля: логин, хэшированный пароль для входа в информационную систему, роль конечного пользователя, логин и пароль для входа в базу данных.

Для авторизации в базе данных в служебной таблице users для каждой роли будет отдельная запись с различными полями привилегий. Это сделано в целях увеличения безопасности пользования системы, в частности для предотвращения атак на MySQL сервер SQL-инъекциями.

В зависимости от роли, находящейся в таблице Authorization, система будет возвращать соответствующую архитектуру главного меню.

Так, роль “Кассир” в главном меню будет иметь функционал в виде продажи билетов и вывода списка пассажиров на конкретный вылет. В базе данных для него в таблице users будет соответствующая запись с необходимыми привилегиями.

Остальные роли будут иметь аналогичную реализацию предоставления прав.

Требования к шаблонам.

* Шаблон сообщений об ошибке (контроллера авторизации).

Для того, что вывести соответствующую ошибку, он должен знать, вследствие чего произошла ошибка: неверная комбинация логина и пароля, либо неправильный логин.

Номер ошибки передаётся с контроллера.

Должен предложить вернуться к статической странице формы авторизации.

* Шаблон формирования списка рейсов (контроллера редактирования списка рейсов).

Должен получать из контроллера PDO Statement объект, который содержит необходимый сформированный двумерный ассоциативный массив, полученный Select - запросом из таблицы Flight, где будут id рейса, аэропорт вылета, аэропорт прилёта;

Должен выводить на экран список рейсов, напротив каждого из них кнопку “Редактировать” и “Удалить”.

Должен знать, что нужно передавать в контроллер при добавлении, редактировании или удалении элемента (id рейса).

* Шаблон формирования списка вылетов (контроллера редактирования списка вылетов).

Должен получать из контроллера PDO Statement объект, который содержит необходимый сформированный двумерный ассоциативный массив, полученный Select - запросом из таблицы Departure, где будут вылета – id вылета, дата вылета, дата прилёта, количество оставшихся билетов и цена за билет.

Должен выводить на экран список вылетов, напротив каждого из них кнопку “Редактировать” и “Удалить”.

Должен знать, что нужно передавать в контроллер при добавлении, редактировании или удалении элемента (id вылета).

* Шаблон формирования списка отчётов.

Должен получать из контроллера PDO Statement объект, который содержит необходимый сформированный двумерный ассоциативный массив, полученный Select - запросом из таблицы Otchet с полями года и месяца, за который был сделан отчёт, количество проданных билетов, суммарная выручка, аэропорт вылета и прилёта.

* Шаблон вывода таблиц из БД.

Должен получать из контроллера PDO Statement объект, который содержит необходимый сформированный двумерный ассоциативный массив, полученный соответствующим запросом.

* Шаблон формирования списка вылетов (контроллера продажи билетов).

Должен получать из контроллера PDO Statement объект, который содержит необходимый сформированный двумерный ассоциативный массив, полученный Select - запросом из таблицы Departure, где будут вылета – id вылета, дата вылета, дата прилёта, количество оставшихся билетов и цена за билет.

Должен выводить на экран список вылетов, напротив каждого из них кнопку “Продать билет”.

Должен передать контроллеру, на какой вылет будет куплен билет (id вылета).

* Шаблон формирования списка вылетов (контроллера списка пассажиров).

Должен получать из контроллера PDO Statement объект, который содержит необходимый сформированный двумерный ассоциативный массив, полученный Select - запросом из таблицы Departure, где будут вылета – id вылета, дата вылета, дата прилёта, количество оставшихся билетов и цена за билет.

Должен выводить на экран список вылетов, напротив каждого из них кнопку “Показать пассажиров”.

Должен передать контроллеру, пассажиров какого вылета нужно вывести (id вылета).

* Шаблон билета.

Должен получать из контроллера PDO Statement объект, который содержит необходимый сформированный ассоциативный массив, полученный Select - запросом из таблицы Ticket, где будут поля id билета, дата вылета, дата прилёта, количество оставшихся билетов и цена за билет.

* Шаблон сообщений об ошибке (общий).

Должен знать номер контроллера, откуда поступила ошибка.

Должен предложить вернуться в главное меню.

Требования к статическим HTML – страницам.

* Статическая HTML – страница формы авторизации.

Должна передавать контроллеру логин и пароль в форме ассоциативного массива.

Имеет поля ввода: логина с атрибутом type = text и пароля с атрибутом type = password.

Так же имеется кнопка “Войти”, которая отправляет запрос на авторизацию контроллеру авторизации, и кнопка “Очистить”, которая очищает поля ввода.

* Статическая HTML – страница формы ввода аэропорта вылета, прилёта и даты.

Должна передавать контроллеру аэропорт вылета, прилёта и даты вылета в форме ассоциативного массива.

Имеет поля ввода: аэропорта вылета и прилёта с тегом select, даты вылета с атрибутом type = date.

Так же имеется кнопка “Отправить”, которая отправляет запрос контроллеру.

* Статическая HTML – страница формы ввода данных пассажира, количества билетов.

Должна передавать контроллеру ФИО пассажира, номер и серию его паспорта в форме ассоциативного массива.

Имеет поля ввода: ФИО пассажира с атрибутом type = text, номера и серии паспорта с аналогичным атрибутом.

Так же имеется кнопка “Отправить”, которая отправляет запрос контроллеру и кнопка “Очистить”, которая очищает поля ввода.

* Статическая HTML – страница формы ввода даты вылета для запроса вывода списка пассажиров.

Должна передавать контроллеру дату вылета в форме ассоциативного массива.

Имеет поле ввода даты вылета с атрибутом type = date.

Так же имеется кнопка “Отправить”, которая отправляет запрос контроллеру.

* Статическая HTML – страница формы выбора между редактированием рейсов и вылетов.

Должна передавать контроллеру информацию о том, какую таблицу редактировать.

Имеет поле выбора категории с атрибутом type = radio.

Так же имеется кнопка “Отправить”, которая отправляет запрос контроллеру.

* Статическая HTML – страница формы ввода добавления/изменения рейсов.

Должна передавать контроллеру id рейса, аэропорт вылета, аэропорт прилёта в форме ассоциативного массива.

Имеет поле выбора аэропорта с тегом select.

Так же имеется кнопка “Отправить”, которая отправляет запрос контроллеру.

* Статическая HTML – страница формы ввода добавления/изменения вылетов.

Должна передавать контроллеру id вылета, дата вылета, дата прилёта, количество оставшихся билетов и цена за билет в форме ассоциативного массива.

Имеет поля ввода: даты вылета и прилёта с атрибутом type = date, количества оставшихся билетов и цены за билет с атрибутом type = text.

Так же имеется кнопка “Отправить”, которая отправляет запрос контроллеру и кнопка “Очистить”, которая очищает поля ввода.

* Статическая HTML – страница формы добавления отчёта.

Должна передавать контроллеру месяц и год, за который создаётся отчёт.

Имеет поля ввода: месяца и года с тегом select.

Так же имеется кнопка “Отправить”, которая отправляет запрос контроллеру.

Логическая модель БД.

Для построения логической модели будущей базы данных применим следующие формальные правила.

а) Каждая сущность инфологической модели отображается отдельной таблицей.

б) Каждая связь типа многие-ко-многим отображается отдельной таблицей.

в) Каждая связь типа один-ко-многим – дополнительным столбцом в дочерней таблице, и этот столбец становится внешним ключом, связанным с первичным ключом соответствующей родительской таблицы.

В результате применения этих правил получим следующий набор реляционных таблиц.

Таблица 1. Рейс – **Flight**

|  |  |  |
| --- | --- | --- |
| f\_id | dep\_airp | arr\_airp |
| Pk |  |  |

Таблица 2. Вылет – **Departure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| d\_id | f\_id | dep\_date | arr\_date | rest\_tickets | price |
| Pk | Fk |  |  |  |  |

Таблица 3. Билет – **Ticket**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| t\_id | c\_id | d\_id | passenger\_name | passenger\_passport | sell\_date |
| Pk | Fk | Fk |  |  |  |

Таблица 4. Кассир – **Cashier**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| c\_id | date\_employment | date\_dismissal | name | passport | birthday |
| Pk |  |  |  |  |  |

Для предотвращения нежелательных изменений в базе данных, созданы четыре группы пользователей: director, technical\_worker, economist и cashier.

Группа director будет иметь возможность только просматривать данные, то есть выполнять SELECT запросы.

Группа technical\_worker помимо права на SELECT запросы, будет иметь права на изменение и удаление данных в таблицах Flight и Departure.

Группа economist помимо права на SELECT запросы, будет иметь права на изменение и удаление данных в таблице Otchet.

Группа cashier помимо права на SELECT запросы, будет иметь права на изменение и удаление данных в таблице Ticket.