

ФИНАЛЬНЫЙ ПРОЕКТ

СОДЕРЖАНИЕ

1	Варианты	2
2	Общие требования	7
3	Требования к оформлению	8
	3.1 Требования к презентации проекта	8
4	Архитектура проекта	9
5	Технологии	10
	5.1 Обязательные к использованию	10
	5.2 Запрещённые к использованию	10
	5.3 Поощряемые	10
6	Список литературы	11

1 Варианты

1. **Система Управления Конференциями.** Существуют роли: Администратор, Модератор, Спикер и обычный Пользователь. Модератор может закреплять, предлагать, изменять тему доклада за спикером, а также регулировать время и место проведения мероприятия. Необходимо учесть возможность просмотра прошедших/будущих митингов. У каждого Спикера существует свой рейтинг, в зависимости от рейтинга начисляется больше бонусов. Спикер может предлагать свой Доклад. Должна быть статистика зарегистрировавшихся людей и сколько физически всего пришли на Доклад. Реализовать уведомление участников о предстоящих событиях.

2. **Система Time-Tracking.** Администратор закрепляет за пользователем Активность. У пользователя может быть одна или несколько Активностей. Пользователь отмечает кол-во затраченного времени на каждую активность. Пользователь может отправить запрос на добавление/удаление Активности.

3. **Система Ремонтное Агенство.** Пользователь может создать заявку на ремонт изделия. Менеджер может принять заявку указав цену, либо отклонить заявку, указав причину. Мастер может выполнить принятую Менеджером заявку. Пользователь может оставить Отзыв о выполненных работах.

4. **Система Кассовый Аппарат.** Кассир может открыть чек, добавить выбранные товары по коду из базы данных (петрушка = 234, хлеб = 222) или по названию товара, указать кол-во товаров или вес. Закрывать чек. Старший кассир может отменить чек, отменить один товар в чеке и вернуть деньги покупателю. Сделать X и Z отчеты. Товаровед может создавать товары и указывать их кол-во на складе.

5. **Система Подачи Отчетов в Налоговую.** Физ./Юр.лицо (далее Пользователь) регистрируется. Подает отчет (XML/JSON/Форма). Налоговый Инспектор принимает/отклоняет отчет (указывая причину отказа). Пользователь может просмотреть все поданные отчеты, причины отказа и изменять их если того потребовал Инспектор. Пользователь может отправлять запрос на замену Инспектора в случае неудовлетворения.

6. **Система Вступительная Компания.** Студент регистрируется на сдачу предметов. Администратор выставляет оценки за предметы. Студент выбирает специальность в университете. Формируются рейтинговые списки поступивших, студентам приходит уведомление о поступлении.

7. **Система Календарь выставок.** Существует список Выставочных залов в каждом из которых есть перечень Экспозиций. Посетитель покупает Билеты оформив Платёж и выбрав Тему выставки.

8. **Система Круизная Компания.** У Компании имеется несколько Кораблей. У Кораблей есть пассажиро-местимость, маршрут, количество посещаемых портов, длительность одного круиза, персонал. Клиент выбирает и оплачивает круиз. Выбирает Экскурсии по прибытии в порт за дополнительную плату. Администратор Корабля указывает бонусы для

пассажиров, учитывая класс билета (бассейн, спорт зал, кинозал, косметические салоны...).

9. Система "Что? Где? Когда?". Система состоит из Знатоков и Судьи. Знатоки отвечают на Вопросы разного типа (с вариантами ответа и без). Судья принимает ответы от Знатоков, опеределяет правильность ответа, и защищает очко команде знатоков (в случае правильного ответа) или команде противников (если Время вышло, или ответ не правильный). Судья также может при запросе от знатоков давать Подсказки разного типа. Сервисы: Время - засекает время между началом и концом вопроса в соответствии с Конфигурацией. Подсказки: • может выводить вероятность правильного ответа (при выборе из нескольких вариантов), • либо убрать несколько неправильных вариантов (при выборе из нескольких вариантов), • либо дать текстовую подсказку (заранее прикреплена к вопросу) • либо дать дополнительное время, и т.п. Статистика - отображает статистику после окончания игры. Формат определяется Конфигурацией Конфигурация - управляет настройками системы: время, количество игроков, тип подсказок, количество вопросов, и т.п.

10. Система Похудения/Трекинга Пищи. Клиент выбирает еду (название, кол. белки, жиры, углеводы), которую съел (из уже готового списка) и пишет Количество. Клиент может добавить свой тип Еды (название, калории, кол. белки, жиры, углеводы). Если Клиент привысил дневную норму, система сообщит ему об этом и напишет, на сколько была превышена Норма. Норму брать из параметров Клиент (возраст, рост, вес, образ жизни и т.д.).

11. Система Автопарк. В систему могут зайти Водители и Администраторы. В Автопарке есть автобусы, маршруты. Администратор может назначать на Маршруты свободные автобусы, в автобусы свободных Водителей, а также освобождать их, делая свободными. Водитель может увидеть свое место работы, а также он должен подтвердить свое новое Назначение.

12. Система Быстрого Тестирования Студентов. Студент регистрируется э-мейлом, к нему привязуется его Успешность и на него будут приходить сообщения о результате тестов. В системе существует каталог Тестов по темам, авторизованный Студент может проходить тесты. В конце теста должна на странице отобразиться форма где показано ошибки студента. Все данные об успеваемости и пройденных курсах отображаются на странице Администратора как сводная таблица по всем Студентам.

13. Система Банковские платежи. Пользователь может иметь один или несколько банковских Счетов (депозитные, кредитные). Доступ к своему Счету можно получить после введения логина и пароля. Пользователь может делать банковские переводы, оплачивать счета, выводить на экран общую информацию (баланса счета, последних операциях, срок действия). Для Кредитных Счетов также доступна информация по кредитном лимите, текущей задолженности, сумме начисленных процентов, кредитной ставкой. Для Депозитных счетов - сумма депозита, ставка, история пополнения.

Пользователь может подать запрос на открытие кредитного счета, если таковой отсутствует. Администратор подтверждает открытие счета с учетом размера депозита и срока действия.

14. Система Доставка Груза. Предоставление услуг по получению и доставке груза. На сайте содержится информация о тарифах и направления доставки. Незарегистрированному пользователю доступен просмотр сайта и калькулятор услуг. Зарегистрированный пользователь у себя в Кабинете может создать Заявку на получение багажа и адрес доставки. Заявка содержит информацию о типе багажа, вес и дату получения. После получения заявки Система формирует счет. Пользователь может оплатить его в своем кабинете.

15. Система Служба Такси. Клиент регистрируется в системе и может создавать заказ на Такси, указывая адрес и тип авто. Программа рассчитывает стоимость поездки в зависимости от дистанции, клиентской скидки и/или акции. Система поддерживает программу Лояльности - рассчитывается скидка в зависимости от суммы потраченной суммы. Также могут действовать акционные предложения. После подтверждения Заказа клиентом, Система ищет свободное такси, соответствующее по типу авто. После чего Клиента оповещают, найдено ли Такси и какое время ожидания.

16. Система Библиотека. Создайте Каталог, по которому можно искать по: • Автору (одному из группы). • Названию книги или её фрагменте. • Одному из ключевых слов книги (атрибут книги). Каталог книг заполняет Администратор, добавляя и изменяя/удаляя их. Каждая книга должна иметь адрес (место на полке) или читателя. Читатель чтобы взять книгу регистрируется, оставляя э-мэйл и номер телефона. Книга может быть взята у Администратора в библиотеке на время не более месяца, только в случае если книга доступна в библиотеке. Администратор должен иметь страницу где отражаются взятые книги и читатели, которые пользуются книгой.

17. Система Расписания Салона Красоты. Нужно создать страницу, где отображается расписание работы сотрудников салона, так чтобы можно было записаться на прием. Записи клиентов должен видеть только Администратор и Мастер, другие видят только занятое место. После предоставления услуг Клиент оставляет отзыв. Предложение о написании отзыва приходит на электронную Почту. Администратор может читать отзывы о коллегах.

18. Система Кинотеатр. Вы пишете интернет витрину маленького Кинотеатра с одним залом. В нем есть Расписание показа фильмов на все 7 дней недели с 9:00 до 22:00 (начало последнего фильма). Незарегистрированный пользователь может видеть: расписание, свободные места в зале, возможность зарегистрироваться. Зарегистрированный пользователь должен быть в состоянии выкупить билет на выбранное место. Администратор может: внести в расписание новый фильм, отменить фильм, просматривать посещаемость зала.

19. Система Периодические издания. Администратор осуществляет ведение каталога периодических Изданий. Читатель может оформить

Подписку, предварительно выбрав периодические Издания из списка. Система подсчитывает сумму для оплаты и регистрирует Платеж.

20. **Система Заказ гостиницы.** Клиент заполняет Заявку, указывая количество мест в номере, класс апартаментов и время пребывания. Администратор просматривает поступившую Заявку, выделяет наиболее подходящий из доступных Номеров, после чего система выставляет Счет Клиенту.

21. **Система Ресторан.** Клиент осуществляет заказ из Меню. Администратор подтверждает Заказ и отправляет его на кухню для исполнения. Администратор выставляет Счет. Клиент производит его оплату.

22. **Система Железнодорожная касса.** Пассажир делает Заявку на билет до необходимой ему станции назначения, время и дату поездки. Система осуществляет поиск подходящего Поезда. Пассажир делает выбор Поезда и получает Счет на оплату. Администратор управляет списком зарегистрированных пассажиров.

23. **Система Больница.** Врач определяет диагноз, делает назначение Пациенту (процедуры, лекарства, операции). Назначение может выполнить Медсестра (процедуры, лекарства) или Врач (любое назначение). Пациент может быть выписан из Больницы, при этом фиксируется окончательный диагноз.

2 Общие требования

Необходимо построить веб-приложение, поддерживающую следующую функциональность:

1. На основе сущностей предметной области создать классы их описывающие.
2. Классы и методы должны иметь отражающую их функциональность названия и должны быть грамотно структурированы по пакетам.
3. Информацию о предметной области хранить в БД. В качестве СУБД рекомендуется MySQL.
4. Приложение должно поддерживать работу с кириллицей (быть многоязычной), в том числе и при хранении информации в БД.
5. Код должен быть документирован.
6. Приложение должно быть покрыто Юнит-тестами
7. События в системе обрабатывать с помощью Log4j.
8. В приложении необходимо реализовать Pagination, Transaction в зависимости от Вашего проекта.
9. Приложение должно корректно реагировать на ошибки и исключения разного рода (Пользователь никогда не должен видеть stack-trace на стороне front-end).
10. В приложении должна быть реализована система Авторизации и Аутентификации.

Servlets:

1. Для доступа к данным использовать API JDBC с использованием пула соединений, стандартного или разработанного самостоятельно.
2. При разработке бизнес логики использовать сессии и фильтры.
3. Используя сервлеты и JSP, реализовать функциональности, предложенные в постановке конкретной задачи.
4. В страницах JSP применять библиотеку JSTL.

Spring:

1. Для доступа к данным использовать Hibernate (JPA).
2. Для фронт-энд части использовать HTML.
3. Реализовать функциональность, предложенную в постановке конкретной задачи, используя Spring Boot.

3 Требования к оформлению

1. Оформление кода должно соответствовать Java Code Convention.
2. Тестовые данные в БД должны быть корректными (не использовать в качестве данных: aaa, bbb, ccc), дабы не возникало проблем при презентации продукта.
3. Проект должен быть залит на *github/bitbucket/gitlab* и иметь соответствующий вид:
 - a. Папка src.
 - b. README.md.
 - c. pom.xml / build.gradle.
4. В проекте должен быть файл README с:
 - a. Описанием проекта (вашего варианта).
 - b. Инструкцией по установке.
 - c. Инструкцией по запуску приложения.

3.1 Требования к презентации проекта

Презентация должна состоять с следующих слайдов:

1. Титульная страница (кто Вы, кто Ваш Тренер).
2. Ваш номер варианта и описание задачи.
3. Требования к проекту.
4. Архитектура проекта (что отвечает за М, V, С).
5. Слайд БД (со связями), к примеру сгенерировать EER Model Вашей БД.

Всё остальные слайды которые Вы хотите от себя добавить – по Вашему желанию, они не являются обязательными.

4 Архитектура проекта (Servlets)

1. Архитектура приложения должна соответствовать шаблону ModelView-Controller.
2. webapp.
3. Сборка приложения происходит при помощи Maven / Gradle (Уметь собирать проект через консоль, не только нажимая зелёную кнопку Run).
4. При реализации алгоритмов бизнес-логики использовать шаблоны:
 - GoF (Factory Method, Command, Builder, Strategy, State, Observer etc.);
 - SOLID – понимать что это и применить в проекте;
 - GRASP – по желанию

5 Технологии

5.1 Обязательные к использованию

1. Log4J;
2. JUnit.

Servlets:

3. JSP + JSTL;
4. Servlets;
5. JDBC.

5.2 Запрещённые к использованию

Servlets:

1. Scriptlets;
2. Applets;
3. Spring / EJB;
4. Hibernate, JPA.

5.3 Поощряемые

1. Mockito;
2. Apache commons;
3. Slf4J;
4. Собственные теги;
5. Собственная Аннотация и Исключения.

6 Список литературы

1. Head First Design Patterns:
<https://www.amazon.com/Head-First-Design-Patterns-BrainFriendly/dp/0596007124>
2. Clean Code, Robert Martin
3. Effective Java:
<https://www.amazon.com/Effective-Java-3rd-Joshua-Bloch/dp/0134685997>
4. Java Методы программирования:
https://careers.epam.by/content/dam/epam/by/book_epam_by/JAVA_Methods_Programming_v2.march2015.pdf