

Лабораторная работа № 1-3. Генерация требований к проектированию пользовательских интерфейсов

Индивидуальные задания (можно выбрать свою тему):

1. Тема: «Автоматизация работы организации по трудоустройству населения».

Разработать программное средство (ПС) «Рынок труда». ПС должно обрабатывать следующую информацию о безработных, зарегистрированных на бирже труда: фамилия, имя, отчество, пол, дата рождения, адрес, образование, учебное заведение, которое закончили, специальность, стаж работы, дополнительные возможности (владение иностранным языком, знание компьютера и т.д.), причина безработицы (сокращение, переезд, болезнь и т.д.). А также информацию о каждом предприятии, предоставляющем работу: название предприятия, адрес, перечень специальностей, имеющих вакансии. Для каждой специальности указаны критерии отбора: образование, стаж, пол, возраст, умения и условия труда на предприятии: рабочий день, выходные, отпуск, заработная плата, льготы и прочее. ПС должно осуществлять подбор для каждого обратившегося безработного наиболее подходящего варианта работы по сведениям в банке данных.

2. Тема: «Автоматизация работы транспортного агентства».

Разработать ПС по автоматизации работы «Транспортного агентства».

ПС должно обрабатывать следующую информацию: по каждому виду транспорта (самолет, поезд, автобус, пароход и т.д.) содержать перечень имеющихся рейсов. Для каждого рейса указывается дата, время отправления и прибытия, пункт назначения и расстояние до пункта назначения, количество посадочных мест, количество проданных билетов, цена билета. Подбирать оптимальный вид транспорта до указанного пункта назначения.

3. Тема: «Автоматизация обработки информации по работе туристической фирмы».

Разработать ПС по автоматизации работы туристической фирмы «Круиз».

ПС должно иметь информацию об отдыхающих: фамилия, имя, отчество, возраст, образование, социальное положение, доход, место (санаторий, база отдыха, дом отдыха, дача и т. д.), время и продолжительность отдыха, сумма, затраченная на отдых.

4. Тема: «Автоматизация учета реализации книжной продукции».

Разработать ПС по автоматизации работ фирмы «Чернокнижник», занимающейся продажей книжной продукции. ПС должно обрабатывать информацию о книгах: форма продажи (почта, магазин, подписка и т.д.), список имеющихся в наличии книг, сгруппированных по тематике (техническая, специальная, художественная и т.д.). При этом для каждой книги указывается серия, код, название, автор, количество экземпляров в наличии, количество проданных экземпляров, цена за единицу. Также ПС должно владеть информацией о заказчиках на книжную продукцию: фамилия, имя, отчество, возраст, адрес, образование, доход, сведения об оплате. Осуществлять подбор литературы по заданной тематике для каждого заказчика.

5. Тема: «Обработка информации по демографической ситуации».

Разработать ПС «Демография», которое должно обрабатывать информацию о новорожденных: фамилия, имя, отчество, пол, дата рождения, место рождения, сведения о родителях. Информацию об умерших: фамилия, имя, отчество, пол, год и место рождения, год и место смерти, причина смерти. Сформировать отчеты о демографической ситуации за определенный период.

6. Тема: «Автоматизация работ культурно-массовых мероприятий».

Разработать ПС, позволяющее автоматизировать работу фирмы «Лира», занимающейся культурно-массовыми мероприятиями. ПС должно обрабатывать следующую информацию: вид культурного заведения (театр, музей, цирк, кинотеатр, дворец культуры и т.д.); для учреждения каждого типа: вид и название проводимого мероприятия (спектакль, кинофильм, выставка, экскурсия и др.), дата и время проведения мероприятия, количество имеющихся и проданных билетов, цена билетов и др., а также анкетные данные для посетителей: возраст, образование, профессия и т.д. Обеспечивать составление программы культурного отдыха по запросу клиента (на определенный день, неделю, месяц); представлять справочную информацию об имеющихся в продаже билетах на определенный спектакль, коллектив, кинофильм, и т.д., позволять бронировать и покупать билеты.

7. Тема: «Учет эмиграции населения».

Разработать ПС, позволяющее обрабатывать информацию по учету эмиграции населения. Данные об эмигрантах могут содержать следующую информацию: возраст, национальность, образование, семейное положение, пол, профессию, страна эмиграции и т.д.

8. Тема: «Социологические опросы населения».

Разработать ПС, позволяющее обрабатывать социологи-ческие опросы населения. Для заполняющих анкету могут использоваться следующие данные: возраст, пол, образование, городской житель или сельский, работающий/безработный и т.д. Опрос может быть проведен на разные темы (политические, социальные и т.д.), предполагается несколько вариантов ответов.

9. Тема: «Анализ работы электронных средств массовой информации».

Разработать ПС, позволяющее проводить анализ работы электронных средств массовой информации (ЭСМИ). Вид ЭСМИ: радио, телевидение, Интернет. Может быть использована следующая информация: сведения об ЭСМИ (вид ЭСМИ, канал, дата, передача, тематика, автор, аннотация, примечания и др.), сведения об авторах (Ф.И.О., вид деятельности (депутат, корреспондент, работник администрации, журналист, активист партии и др.).

10. Тема: «Анализ информации о содержании и воспитании детей в детском доме».

На каждого ребенка, поступающего в детский дом, заводится паспорт или личное дело, в котором хранится информация о месте рождения ребенка, его возраст, откуда

поступил в детский дом и по каким причинам, а также данные о получаемых пособиях, пенсиях, алиментах. Ребенок может поступать в детский дом из другого детского дома, при отказе от ребенка, при лишении родительских прав и т.п. В паспорте хранятся данные о номере и дате распоряжения о направлении и номере и дате путевки в детский дом. За ребенком, который поступил в детский дом, может быть закреплен жилой дом или квартира. Адрес этого жилья также вносится в личное дело ребенка. В личном деле хранится информация, куда он отправлен после выхода из детского дома (направление). В личном деле хранится информация о состоянии здоровья ребенка (прививки, профилактические мероприятия). Если ребенок поступает в детский дом с медицинской картой, то данные о прививках берутся из нее, если прививки делаются в детском доме, то хранится дата, характер и результат прививки.

11. Тема «АРМ Библиотекаря».

Автоматизированная система предназначена для автоматизации работы библиотекаря учебного заведения. БД должна вести учет:

- а) Справочника книг, разбитых по тематике (Добавление, удаление, редактирование информации о книге);
- б) Каждого экземпляра, находящегося в библиотеке (у каждого переплета должен быть собственный номер, шифр, обозначающий положение книги в хранилище);
- в) Книг, подлежащих списанию (определить условие для списания книг, при списании обязательно указывать причину);
- г) Читателей библиотеки (добавление, удаление, редактирования информации о читателе);
- д) Карточек читателей (выдавать карточку читателей – список книг, заказываемых читателем, дату заказа книги, дату возврата, выдавать список читателей, вовремя не сдавших книги, информировать читателей, начислять штрафные санкции);
- е) Поставщиков книг (добавлять, редактировать и удалять информацию о поставщиках, у которых колледж заказывает книги)
- ж) Поставок книг (когда и на какую сумму были произведены закупки книг)

12. Тема «АРМ работника склада».

Автоматизированная система предназначена для автоматизации работы сотрудника склада. БД должна вести учет:

- а) Заказчиков (добавление, редактирование и удаление информации обо всех заказчиках данной организации);
- б) Заказов (когда, по какой цене, кем и кому были отгружены товары);
- в) Продукции (какие товары, по какой цене, в каком количестве находятся на складе, формировать отпускную цену);
- г) Поставок (когда, кем, в каком количестве и по какой цене были произведены поставки товаров);
- д) Поставщиков (добавление, удаление и редактирование информации о всех поставщиках продукции);
- е) Работников склада (учет грузчиков, начисление им зарплаты);
- ж) Прибыли склада (по какой цене закуплена продукция, по какой цене продана продукция, стоимость суток хранения единицы продукции).

13. Тема «АРМ администратора ателье по ремонту оргтехники».

Автоматизированная система должна вести учет:

- а) Клиентов ателье (добавление, удаление и редактирование информации обо всех клиентах ателье)
- б) Техники, сданной в ремонт (какая техника, когда сдана в ремонт, какой ремонт необходим, в какой срок выполнится, степень готовности и информация о мастере, который будет выполнять работу)
- в) Комплектующих (стоимость комплектующих, использованных при ремонте, отдельный заказ комплектующих);
- г) Прделанной работы (какая техника, стоимость работы по категориям, формирование общей стоимости заказа, учет скидок и налогов при формировании цены, вид оплаты)
- д) Работников ателье (формирование заработной платы работникам, в зависимости от объема выполненных работ).

14. Тема «АРМ администратора ресторана».

Автоматизированная система должна вести учет:

- а) Распределения столиков (на сколько мест, где расположены, учитывать, какая наценка идет на столик, в зависимости от его расположения);
- б) Клиентов ресторана (добавление, удаление и редактирование информации обо всех клиентах ресторана);
- в) Предварительных заказов на столики (распределение предварительных заказов на столики, предоплата, меню на столик);
- г) Меню (формирование меню на конкретный день, стоимость блюд, напитков);
- д) Заказов на конкретный столик (формирование счета в зависимости от заказанных блюд, принятие оплаты).

15. Тема «Автоматизация учебного процесса».

Автоматизированная система должна вести учет:

- а) Всех кафедр колледжа (добавление, удаление и редактирование информации обо всех кафедрах колледжа);
- б) Всех преподавателей колледжа (добавление, удаление и редактирование информации обо всех преподавателях колледжа);
- в) Всех групп колледжа (добавление, удаление и редактирование информации обо всех группах и отделениях колледжа);
- г) Всех студентов колледжа (добавление, удаление и редактирование информации обо всех студентах колледжа);
- д) Успеваемости студентов колледжа (информация о предмете, преподавателе, номере семестра, виде занятия (зачет, экзамен), группе, студенте и итоговой оценке);
- е) Изучения предметов (в какой группе, какой преподаватель и какой предмет читает с указанием часов на практические и лабораторные работы).

Порядок выполнения работы

1 Составить документ «Профиль клиента», в котором кратко описать информацию о компании (1...2 страницы).

2 Разработать видение ИС (видение выполнения проекта и границы проекта – документ, который кратко описывает, в каких подразделениях и в какой функциональности будет внедряться ИС), в том числе:

проанализировать проблемную ситуацию, определить позицию разрабатываемой АИС;

сформулировать краткое описание АИС, определить его возможности; выработать и описать прочие требования к АИС.

Шаблон документа «Видение» содержит следующие основные разделы.

1 Введение.

2 Позиционирование.

3 Описания совладельцев и пользователей.

4 Краткий обзор изделия.

5 Возможности продукта.

6 Ограничения.

7 Показатели качества.

8 Старшинство и приоритеты.

9 Другие требования к изделию.

10 Требования к документации.

11 Приложение.

Во введении описываются цель документа, его контекст (связь и взаимовлияние с различными проектами), определения, сокращения, ссылки на другие документы, краткое содержание.

В разделе «Позиционирование» помещается определение решаемой проблемы (проблем), указывается целевой заказчик и исследуются деловые преимущества изделия перед аналогичными на рынке.

В описании совладельцев и пользователей, помимо собственно описания этих двух групп, исследуется демография рынка: целевые рыночные сегменты, размер и темпы роста рынка, существующие конкурентные предложения на рынке, репутация Разработчика на рынке.

Краткий обзор изделий содержит резюме изделия, описание его перспектив и ключевых возможностей, предположения и зависимости, указывается стоимость и ее калькуляция, рассматриваются вопросы лицензирования и инсталляции.

В разделе, посвященном возможностям продукта, они описываются более подробно, каждая – в отдельном параграфе.

В раздел «Ограничения» следует выносить существующие технические, технологические и др. обстоятельства, которые необходимо учитывать на данной стадии.

Раздел «Показатели качества» содержит описание наиболее существенных нефункциональных требований к системе (эффективности, надежности, отказоустойчивости и др.).

Раздел «Старшинство и приоритеты» ранжирует сформулированные ранее требования и возможности системы по степени важности, очередности реализации и т. п.

Раздел «Другие требования к изделию» описывает применяемые стандарты, системные требования, эксплуатационные требования, требования к окружающей среде.

В требованиях к документации приводятся ключевые характеристики руководства пользователя, интерактивной справки, руководства по установке и конфигурированию, файла Read Me.

В приложение выносятся атрибуты возможностей. RUP рекомендует следующий набор атрибутов: статус, выгода, объем работ, риск, стабильность, целевой выпуск, назначение, причина.

3 Оформить работу.

4 Осуществить защиту работы.

Лабораторная работа № 4-6. Разработка технического задания к проектированию пользовательских интерфейсов

Порядок выполнения работы

1. Изучить пример технического задания.
2. Составить документ «Техническое задание».
3. Оформить работу.
4. Осуществить защиту работы.

Пример технического задания

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на создание

системы проведения мероприятий Инновационного центра «Сколково» для нужд Некоммерческой организацией Фонд развития Центра разработки и коммерциализации новых технологий

СПИСОК ПРИНЯТЫХ СОКРАЩЕНИЙ

API	Программный интерфейс для интеграции с внутренними и внешними системами
Backend	Основная программно-аппаратная часть системы
CMS	Информационная система, используемая для обеспечения и организации совместного процесса создания, редактирования и управления контентом
Event-платформа	Внешняя система, предоставляющая функционал организации мероприятий
GUI	Графический пользовательский интерфейс — визуальное представление системы
REST API сервис	Веб-сервис, который предоставляет доступ к программному интерфейсу системы по протоколу HTTPS
Администратор	Пользователь с правами объект-менеджера, картографа, модератора и возможностью редактировать роли и функционал других пользователей
База данных системы	Внутренняя база данных системы
Журнал операций	База данных, в которой хранятся записи об определенных операциях для дальнейшего анализа и составления отчетов
Интерфейс администратора	Веб-представление административной панели
Интерфейс служебного пользователя	Веб-представление административной панели, ограниченного правами пользователя, функционала.
ИЦС	Инновационный центр «Сколково»
Пользователь	Человек, взаимодействующий с системой в качестве потребителя услуг и информации

1. ЗАДАНИЕ НА РАЗРАБОТКУ

Создание Системы осуществляется путем доработки существующей технологической платформы сайта мероприятий, принадлежащей Фонду Сколково и разработки новой функциональности, в соответствии с Функциональными требованиями, описанными ниже.

Исходный код существующей системы предоставляется Фондом.

Технологический стек:

На серверах баз данных используются:

- операционная система Debian 7;
- программно-аппаратный элемент Firewall на iptables и fail2ban;
- СУБД PostgreSQL 9.

На серверах веб-приложения используются:

- операционная система Debian 7;
- программно-аппаратный элемент Firewall на iptables и fail2ban;
- PHP 7 с расширениями GD, mcrypt и pgsql;
- nginx 1.13 и php-fpm.

2. ОБЩИЕ ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Задачи системы:

- помогать Фонду Сколково и его ДО продвигать собственные и партнерские активности, мероприятия;
- наполнять экосистемы Фонда Сколково и его ДО социокультурными мероприятиями;
- привлекать внешних заказчиков для проведения мероприятий;
- продвигать Фонд Сколково и его ДО как современный культовый событийный центр для инноваторов и технологических предпринимателей;

Система должна:

- предоставлять полную информацию о мероприятиях и событиях, проходящих в Фонде Сколково и его ДО, время и место проведения, программу, информацию о спикерах и участниках;
- позволять гостям регистрироваться, оплачивать участие в мероприятии, составлять план посещения сессий, назначать встречи, задавать интересующие вопросы докладчикам, давать обратную связь по итогам проведения мероприятия;
- предоставлять организаторам мероприятий комплексный сервис проведения мероприятий, включая привлечение участников, конструктор промостраниц мероприятий, возможность онлайн-регистрации и оплаты участия, управления программой;
- предоставлять организаторам мероприятий статистику по проведенному мероприятию, информацию по опросам участников, обратную связь от участников;
- предоставлять администраторам мероприятий единую систему управления организацией мероприятий, включая управление конгресс-центром, его доступностью и утилизацией;
- оперативно создавать сайты мероприятий, предоставляющие полный спектр услуг по регистрации и оплате, назначения встреч, печати билетов, управления программой
- централизованно управлять пользователями, проводить анкетирование, аналитику интересов, таргетированно информировать о событиях в Фонде Сколково и его ДО;
- осуществлять рассылки по пользователям с применением различных фильтров выборки получателей.

3. ОПИСАНИЕ СИСТЕМЫ

3.1 Пользовательские роли

Система должна поддерживать следующую ролевую модель, а также возможность управления ролями и их правами.

Основные ролевые группы и их состав:

3.1.1. Пользователи сайта:

- а) Незарегистрированный пользователь;
- б) Зарегистрированный пользователь.

3.1.2. пользователи CMS:

- a) Организатор мероприятия;
- b) Администратор системы.

3.1.1 Пользователи сайта

Эта ролевая группа включает в себя пользователей системы, обладающих первичным ограниченным доступом.

a) **Незарегистрированный Пользователь** – это пользователь, который не авторизовался в системе.

Пользователь получает актуальную запрашиваемую информацию в режиме реального времени.

Пользователь может:

- просматривать текстовые информационные страницы;
- просматривать списки мероприятий;
- просматривать карточки и сайты мероприятий;
- осуществлять взаимодействие с формами регистрации и авторизации.

b) **Зарегистрированный пользователь** – это пользователь, который имеет учетную запись и успешно произвёл авторизацию в системе.

Зарегистрированный пользователь наследует права и возможности незарегистрированного.

Зарегистрированный пользователь может:

- регистрироваться на мероприятия, оплачивать участие;
- пользоваться сервисом деловых встреч;
- задавать вопросы докладчикам;
- давать обратную связь по итогам мероприятия;
- получать персональные уведомления;
- восстанавливать пароль через емейл;
- редактировать данные профиля.

3.1.2 Пользователи CMS

Эта ролевая группа включает в себя системных пользователей, которые имеют ограниченное и распределённое управление системой и её контентом.

Все пользователи этой группы наследуют права и возможности зарегистрированного пользователя.

3.1.1 Организатор мероприятия

Организатор мероприятия может:

- добавлять и редактировать название, логотип, категорию, место, дату начала, краткую информацию, программу, спикеров, детальную информацию;
- добавлять фотографии и видео в фотогалерею;

Информация, добавленная организатором мероприятия может проходить обязательную проверку администрацией.

3.1.2 Администратор

Администратор может:

- проверять информацию, созданную или изменённую организатором мероприятия;
- размещать оповещения и мероприятия от имени Фонда Сколково и его ДО;
- выгружать статистику и активность посетителей мероприятий.

3.2 Компоненты системы

Предварительно считается, что система будет состоять из следующих компонентов:

3.2.1 Backend

Backend – это та серверная часть системы, внутри которой происходит вся бизнес логика и процессы.

3.2.2 REST API Сервис

Взаимосвязь и отношения с Backend происходят посредством обращения клиентских приложений к REST API сервису. Интеграция со сторонними системами так же производится посредством API сервиса.

3.2.3 Веб-приложение

Веб-приложение — это клиентская часть системы, для взаимодействия с конечными пользователями. Способ потребления и производства информации пользователем, посредством взаимодействия с веб-сайтом навигационной системы.

Сайт и административная часть системы является независимыми компонентами веб-приложения.

3.3 Система управления мероприятиям

Система управления мероприятиями обеспечивает функции создания, управления и поддержки мероприятий.

3.3.1 Свойства мероприятия

Мероприятие должно иметь возможность содержать следующие свойства:

- Название мероприятия строка
- Логотип файл форматов jpeg, gif, png
- Категория
- Город выпадающий список (варианты подбираются по мере набора в текстовом поле)
- Адрес строка (используется сервис «Дадата»)
- Дата начала строка формата dd.mm.yyyy
- Дата окончания строка формата dd.mm.yyyy
- Сайт мероприятия строка, включая протокол, если протокол не введен, то по умолчанию подставляется http://
- Краткая информация текст (подсказки возле лейблов будут указывать на то, где будет отображаться данная информация)
- Детальная информация текст(подсказки возле лейблов будут указывать на то, где будет отображаться данная информация)
- Спикеры мероприятия — список выступающих на мероприятии.

Модуль мероприятий позволяет создавать промостраницу для мероприятия: добавлять логотип, тексты в шапку, подвал и основную часть. Размещать описание, фото и видео, спикеров, программу мероприятия. Промостраница доступна по персональной ссылке всем, включая незарегистрированных пользователей и при необходимости может быть закреплена за отдельным доменом второго уровня. В зависимости от настроек мероприятия, промостраница может содержать форму регистрации.

Модуль управления программой мероприятия, позволяющий через административный интерфейс управлять информацией о залах, временными слотами и секциями.

Управление программой обеспечивает следующие возможности:

- Формирование программной сетки (привязка секций к временным слотам)
- Управление залами/площадками проведения программных мероприятий
- Редактирование информации по каждой секции:
 - Название
 - Дата и время проведения
 - Описание
 - Ведущие, спикеры, модераторы
 - Тезисы и материалы

В публичной части есть функционал генерации программной сетки на основе введенных данных. Программная сетка состоит из общей карты программы (краткая информация о секциях), а также страниц с детальной информацией о каждой секции.

Модуль программы подключается к мероприятиям, а также работает с имеющимися сущностями пользователей.

У мероприятия есть форма регистрации, список участников со статусами участия.

3.3 Система управления пользователями

Система управления пользователями позволяет пользователям регистрироваться в системе через сервис SK.ru, редактировать свой профиль, оплачивать мероприятия, принимать в них участие, получать бейджи, дипломы и сертификаты.

Регистрация и авторизация пользователя осуществляется через сервис SK.ru

Личный кабинет пользователя содержит следующие поля, обязательность заполнения которых определяется в настройках каждого конкретного мероприятия:

- Внутренний ID Числовой тип
- Фамилия Строковый тип
- Имя Строковый тип
- Отчество Строковый тип
- Компания Строковый тип
- Должность Строковый тип
- Email Строковый тип
- Пароль Строковый тип (не менее 6 символов)
- Пол Радиогруппа (М/Ж)
- Дата рождения Строковый типа формата dd-mm-уууу
- Телефон Строковый тип формата +(код)-(номер) (Несмотря на то, что тип строковый, допускаются только числовые символы [0-9])
- Фото файл форматов: jpg, png, gif
- Гражданство/проживание в стране Радиогруппа (Россия/Да, Россия/Нет, Иностранное государство/Да) (для ФСО)
- Данные паспорта РФ (для ФСО)
- согласия с правилами пользования и обработкой персональных данных – Чекбокс

Интерфейс управления участниками мероприятия для организаторов должен позволять:

- Просматривать список зарегистрированных участников;
- Осуществлять поиск и фильтрацию среди участников по ФИО, контактной информации и статусу на мероприятии;
- Позволять редактировать информацию об участии пользователя в мероприятии как в индивидуальном, так и массовом режиме;
- Иметь механизмы автоматизированной выгрузки участников в структурированном формате (Excel или CSV).

На уровне модуля регистрации участников на мероприятия должна быть предусмотрена подключаемая возможность автоматического создания электронного билета на мероприятие, содержащего информацию о названии, дате/времени и месте проведения события, а также данные зарегистрированного участника и QR-код для быстрой идентификации на площадке (при наличии соответствующего АПК на площадке).

3.4 Деловые встречи

Система позволяет пользователю создать и посетить деловую встречу на территории ИЦ Сколково.

Есть определённое количество переговорных зон. Пользователь может забронировать такую зону на определённый день и время, пригласить коллег, которые зарегистрированы в общей базе CMS.

Функционал модуля деловых встреч должен позволять:

- Осуществлять поиск интересующих пользователей с возможностью применения поиска/фильтрации по полям анкеты среди зарегистрированных участников на конкретное мероприятие;
- Просматривать список назначенных встреч, видеть их статус и иметь возможность управлять им для каждой из встреч (подтверждать, отклонять);
- Поддерживать режим уведомлений по электронной почте (в случае назначения встречи или изменения ее статуса);

- Иметь режим автоматического бронирования переговорных комнат для участников встречи в случае ее подтверждения обоими сторонами;
- Отображать для организаторов мероприятий статистическую информацию по встречам, список всех заявок на встречи с указанием статуса и возможности быстрого поиска/фильтрации по различным характеристикам.

3.5 Финансы модуль мероприятий

Модуль учета финансовой аналитики позволяет организовать приём оплат за участие в мероприятиях для физических и юридических лиц.

Способы оплаты:

- Физические лица
 - Банковские карты
 - Электронные деньги
 - Квитанции для оплаты в банке
- Юридические лица
 - Выставление счетов
 - Оплата корп. картами

Процедура оплаты и учета поступлений для физических лиц полностью автоматизируется. Для обработки платежей осуществляется подключение к эквайринговой компании, такой как Cloudpayments, PayOnline, Uniteller, Chronopay и т.д.

Для юридических лиц автоматизируется процедура подготовки платежных документов, проставление отметок осуществляется в ручном режиме через специальный интерфейс.

Должна быть предусмотрена возможность генерации промокодов для конкретных мероприятий с возможностью задания следующих параметров при создании:

- Количество кодов
- Срок действия
- Размер скидки
- Количество активаций

Модуль имеет собственный интерфейс, позволяющий предоставить доступ к финансовой информации, аналитике.

Должна быть обеспечена интеграция со следующими системами:

- с учётной системой на платформе 1С;
- с системой биллинга;
- с системой CRM;

Окончательный перечень систем, подлежащих интеграции, и требования к интерфейсам интеграции определить совместно с Заказчиком при детализации Технического задания.

3.6 Электронные рассылки

Модуль должен позволять организаторам мероприятий и администраторам осуществлять рассылки по базе участников.

Организаторы мероприятий должны иметь доступ только к базе участников подотчетного им мероприятия. Каждая рассылка перед отправкой должна быть проведена и одобрена со стороны администратора.

Администраторы должны иметь возможность осуществлять рассылки по всей пользовательской базе участников.

В модуле должна быть предусмотрена возможность гибкого определения таргетинга получателей (по мероприятиям в которых участник принимал участие, по региону, полу). Также должна быть предусмотрена возможность тестовой отправки писем.

Функционал модуля рассылок должен предусматривать возможность создания ссылок «быстрой авторизации», то есть позволяющей автоматически идентифицировать и авторизовать пользователя при переходе по специальной ссылке из рассылки.

3.7 Статистика

Модуль статистики позволяет получать статистику по посещению всех мероприятий или конкретного мероприятия в следующих разрезах:

- Пол
- Возрастные группы
- Смежные интересы
- География участников
- Сфера деятельности

Модуль позволяет отслеживать статистику активности конкретного пользователя на протяжении времени:

- Какие мероприятия посетил
- Какая конверсия из подписчика в посетителя
- Рост социального графа

3.8 Поиск и фильтрация

Система предоставляет возможность поиска по мероприятиям.

Критерии поиска и фильтрации:

- по категории,
- по краткому описанию,
- по полному описанию,
- по названию.

3.9 Отображение полезной информации

Система может отображать следующий тип информации:

- информационные страницы;
- данные по мероприятиям: списки и карточки;
- данные по участникам и спикерам мероприятий.

4. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К СИСТЕМАМ

4.1 Система управления личными данными пользователей

Система управления пользователями должна:

- позволять заполнять и редактировать личные данные;
- позволять пользователю регистрироваться на мероприятия;
- позволять пользователю получать документы о посещении мероприятия

4.2. Система управления мероприятиями

Система управления мероприятиями должна:

- позволять организаторам создавать и редактировать мероприятия;
- позволять организаторам создавать промостраницу мероприятия;
- позволять организаторам открывать и закрывать регистрацию, печатать билеты, бейджи и сертификаты и дипломы;
- позволять администраторам проверять и анонсировать мероприятия;
- позволять пользователям создавать личную программу в кабинете;
- позволять пользователям задавать вопросы докладчику.

4.3 Интерфейсы

Веб-приложение

В данном разделе описаны основные требования к веб-приложению.

Общие требования к веб-приложению

Приложение должно отвечать общим функциональным требованиям, представленным в разделе «Общие функциональные требования».

Административная часть

Доступ к административной части системы осуществляется посредством авторизации пользователя с соответствующей ролью.

Пользователь административной части системы должен иметь возможность управлять настройкой и/или производить администрирование системы без дополнительных технических навыков посредством GUI веб-приложения.

Административная часть системы должна быть простой и удобной в использовании с сопровождением подсказок и справок, составленных разработчиком ПО.

В административной части имеются отдельные разделы для управления:

- Мероприятиями:
 - функционал идентичен с ролью пользователя «Организатор мероприятия».
- Пользователями, группами пользователей и правами доступа.
- Контентом, представленным в системе.
- Сбором данных (отчёты и статистика).
- Просмотром статистики (составление отчётов и графиков).

4.4 Сбор персональных данных

Система должна собирать персональные пользовательские данные с разрешения пользователя, такие как:

- поисковые запросы;
- частота пользования системой;
- сбор геопозиционных данных при их доступности.

5. ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

5.1 Требования к платформе

Backend системы должен:

- быть написан на языке последнего стабильного релиза, оптимизированном под большие нагрузки и работу с виртуализацией; на усмотрение исполнителя, но с одобрением заказчика;
- использовать парадигму Микро-сервисов и иметь продуманный механизм развёртывания (Deployment) с использованием современной платформы «Docker» или другого гибкого, быстрого и актуального механизма;
- быть написан по всем канонам современной разработки с использованием современных библиотек, с приоритетом на «открытый код»;
- учитывать, что максимальное критическое время на отображение конечного результата не превышает 500 мс для обычных операций, таких как отображение карты, объектов и вывод информации по объектам и не более 5 секунд для сложных расчётов, таких как формирование отчётов;
- быть компонентным, расширяемым;
- быть масштабируемым (кластерной, учитывающая серверную виртуализацию).

5.2 Требования к GUI мобильного приложения

GUI приложения должен учитывать рекомендации к проектированию интерфейса от производителя операционной системы.

GUI должен отвечать требованиям фирменного брендбука и должен быть согласован с заказчиком.

5.3 Требования к API системы

Система подразумевает под собой набор конечных точек API, с помощью которых возможно

использование её функционала. Работа системы должна быть устроена так, чтобы была возможность добавлять новые интерфейсы (клиентские части) без вмешательств в программный код самой системы и ядра в целом.

API системы должен:

- отвечать всем канонам современной разработки (HTTPS, WebSockets, RESTful, OAuth и т.д.);
- быть максимально полным и покрывать весь функционал пользовательской и административной части системы. Работа всех клиентских частей должна производиться посредством использования этого API;
- сопровождаться всеобъемлющей документацией с примерами кода, в закрытом доступе;
- предоставлять систему авторизации сторонних приложений (например, token). Выдача доступа осуществляется администрацией.

5.4 Требования к информационной безопасности

- Используемая платформа и стороннее ПО должно соответствовать стандарту информационной безопасности ISO/IEC 27001 и гарантировать безопасность общих и пользовательских данных.
- Уровень защиты персональных данных должен отвечать требованиям Ф3-№152.
- Для выявления наличия уязвимостей разрабатываемой системы, Заказчик вправе проводить аудит системы в целом или отдельных ее элементов. Условия и сроки проведения таких аудитов более подробно описаны в Техническом задании к конкурсной документации.

6. ВНЕШНИЕ СЕРВИСЫ

Система должна поддерживать внешние сервисы, такие как:

- сервис почтовых рассылок;
- провайдер авторизации пользователя SK.ru

7. ПЛАН ГРАФИК РАБОТ

Работы выполняются и передаются заказчику поэтапно в соответствие с планом графиком.

№	Вид работ / модуль	Описание	Дата запуска
1 ЭТАП РАБОТ			
1	Разработка дизайн-концепции публичной части	Дизайн-макет главной страницы и основных внутренних разделов, а также минимально-необходимых системных интерфейсов	15 декабря 2017
2	Проектирование системы, подготовка ядра	Разработка архитектуры АИС, проектирование базы данных, подготовка и сборка ядра	15 декабря 2017
3	Интерфейс управления мероприятием (для организаторов)	Специальный интерфейс для организаторов события, отображающий информацию по конкретному мероприятию: статистику, списки участников, настройки, финансовые разделы (после реализации соответствующего модуля), выгрузки и т.д.	15 декабря 2017
2 ЭТАП РАБОТ			

1	Основные публичные интерфейсы (разделы)	Главная страница, пользовательские профили, статичные текстовые общеинформационные разделы, форма восстановления пароля, интерфейсы авторизации и регистрации	1 апреля 2018
2	Административный интерфейс (для Сколково)	Разработка панели управления для администраторов системы, позволяющей: просматривать/редактировать списки пользователей и мероприятий, осуществлять настройки и мониторинг, обеспечивать доступ к управлению другими модулями	1 апреля 2018
3	Публичный календарь мероприятий	Наглядное представление всех запланированных и прошедших мероприятий с возможностью применения фильтров по различным критериям и экспортом в календарь Outlook / Google Calendar	1 апреля 2018
4	Публичная настраиваемая страница мероприятия	Публичная страница мероприятия, содержащая элементы брендинга мероприятия, название и даты проведения, описание, форму регистрации, контакты и другую информацию добавленную через интерфейс управления мероприятием	1 апреля 2018
5	Публичная настраиваемая регистрация мероприятия на	Возможность производить гибкую настройку формы регистрации на мероприятие, указывая дополнительные поля, их тип, обязательность заполнения, порядок отображения	1 апреля 2018
6	Подсистема интеграции с sk.ru	Возможность авторизации в системе через аккаунт в sk.ru (через OAuth)	1 апреля 2018
7	Интерфейсы для партнеров и экспонентов	Специальные интерфейсы для партнеров и экспонентов мероприятия, позволяющие им самостоятельно управлять информацией, связанной с их компанией	1 апреля 2018
8	Сервис деловых встреч	Возможность подключения модуля деловых встреч (упрощенный, по сравнению с "Открытыми инновациями") для конкретных мероприятий	1 апреля 2018
9	Набор виджетов для интеграции регистрации с внешними сайтами	Автоматическая генерация HTML/JS кода для интеграции регистрационной формы на другие ресурсы	1 апреля 2018

10	Модуль финансового сопровождения	Комплекс механик и интерфейсов для обеспечения процесса финансового сопровождения: интеграция с эквайером, выставление счетов, интерфейс просмотра платежей и проставления оплат, применение промо-кодов	1 апреля 2018
11	Модуль управления и автоматического формирования программной сетки	Интерфейс формирования списка секций во временной сетки и привязки к ним зарегистрированных спикеров	1 апреля 2018
12	Модуль информационных рассылок с настраиваемым таргетингом	Административный модуль, позволяющий отправлять форматированные тексты по пользовательской базе с применением различного таргетинга	1 апреля 2018
13	Модуль аналитики по мероприятиям (для Сколково)	Табличное и графическое представление ключевых показателей по пользователям и мероприятиям	1 апреля 2018
14	Модуль интеграции с системой аккредитации	Возможность двусторонней интеграции с ПО по осуществлению аккредитации участников на площадке	1 апреля 2018

Лабораторная работа № 7 – 8 Построение use case диаграмм и диаграмм деятельности.

1 .Методические указания

Карты элементов use case

Элементы use case не существуют отдельно от внешнего мира. Полноценная программная система должна обеспечивать поддержку десятков, а то и сотен элементов use case, причем, внутри каким-то образом связанного с ней приложения должна существовать связь между этими элементами. Отображение взаимосвязи между приложениями дает возможность описать общую структуру задачи, решаемой приложением и его интерфейсом. Карта элементов use case для данной задачи разбивает все функциональные возможности системы на множество взаимосвязанных сущностных элементов use case. Выделив все различающиеся и важные взаимодействия и показав отношения между ними, можно создать упрощенную общую модель задач, решаемых системой, и возможностей, которые она обязана предоставить.

Полноценная модель use case представляет собой множество описаний, определяющих суть всех элементов use case, и карту этих элементов, показывающую отношения между ними. Между сущностными элементами use case могут существовать отношения разных типов, включая специализацию, расширение, композицию, а также сходство. Знание этих отношений позволяет аналитику или разработчику выделить общие элементы задачи и в итоге создать более простую модель задачи.

Специализация

Некоторые элементы use case могут являться специализированными версиями других элементов. Например, при разработке приложения «банкомат» элементы use case «получениеДенег», «размещениеСредств» и «запросСостояния» являются субклассами, или специализированными вариантами абстрактного класса взаимодействий, который может быть назван «использованиеБанкомата». Что касается отношения между элементами «получениеДенег» и «использованиеБанкомата», то его можно охарактеризовать как классификацию, или специализацию. Такой тип отношения означает, что один элемент use case «является» («is-a») специализацией другого. В объектно-ориентированном анализе и проектировании такое отношение соответствует отношению класс-подкласс.

Специализация дает возможность упростить общую модель use case путем отделения общих или универсальных форм взаимодействия от специфических форм, адаптированных для более узкого применения. Таким образом, нет необходимости переписывать заново самые общие паттерны применения. Достаточно написать их один раз, а затем лишь «повторно использовать» («reuse»), ссылаясь на них. Как видно из рис. 1, для отображения отношения специализации используется двойная стрелка. Рядом со стрелкой можно встретить подпись «is-a» или «specialize», в зависимости от контекста.

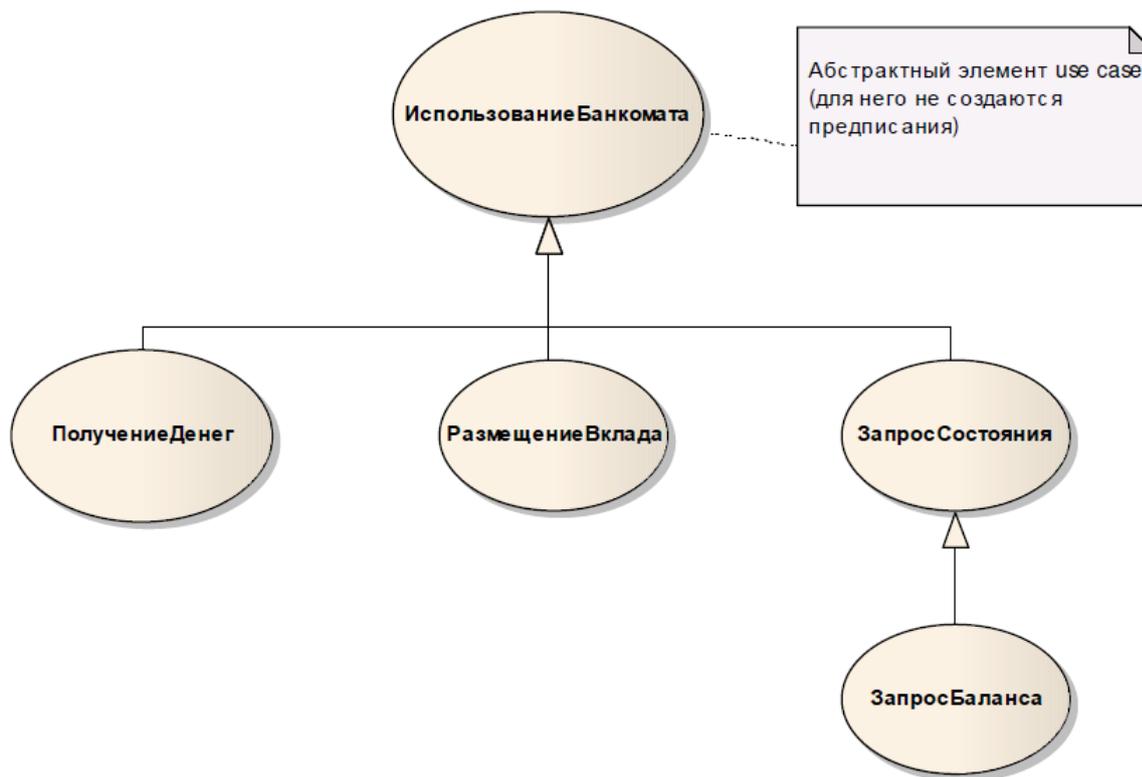


Рис. 1. Отношения специализации на карте элементов use case

Расширение

Одной из инноваций в объектно-ориентированной программной инженерии, навеянных идеями Якобсона, стало признание расширения одним из возможных отношений между элементами use case. Говорят, что один элемент «расширяет» другой, когда он содержит вставляемые или альтернативные паттерны взаимодействия, которые войдут в расширяемый элемент. Например, при отработке элемента use case для изменения внешнего вида какой-то части экрана пользователю необходимо заниматься поиском по всей системе файла, содержащего нужную картинку или значок. Нормальное, или ожидаемое, развитие событий (обеспечивается базисом, или базисным элементом use case) тем не менее вовсе не подразумевает поиск каких-то дополнительных графических файлов.

Расширение – это удачная концепция, позволяющая значительно упростить сущностные модели use case. На карте элементов use case отношение расширения изображается в виде пунктирной линии со стрелкой и имеет подпись «extend», как показано на рис. 2. Если для расширения дается дополнительное описание, в него может быть включено примечание, показывающее, какие элементы use case расширяются.

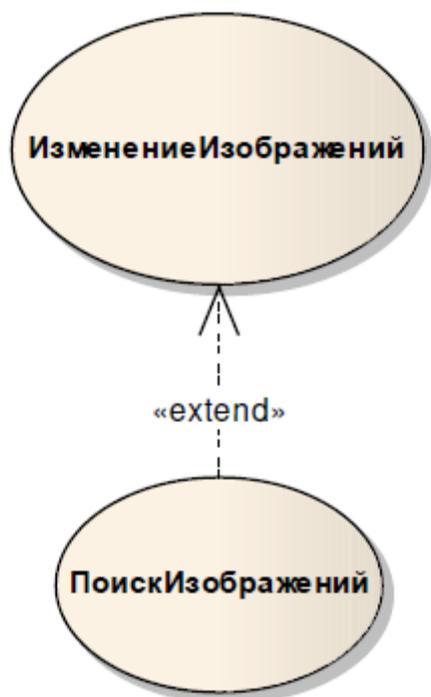


Рис. 2. Отношение расширения

Композиция

Элементы use case можно декомпозировать на составные части, или подэлементы, являющиеся подчиненными или включенными паттернами взаимодействия. Отношение композиции обозначается на карте элементов use case пунктирной стрелкой, указывающей на подэлемент use case и имеющей метку «include» как показано на рис. 3.

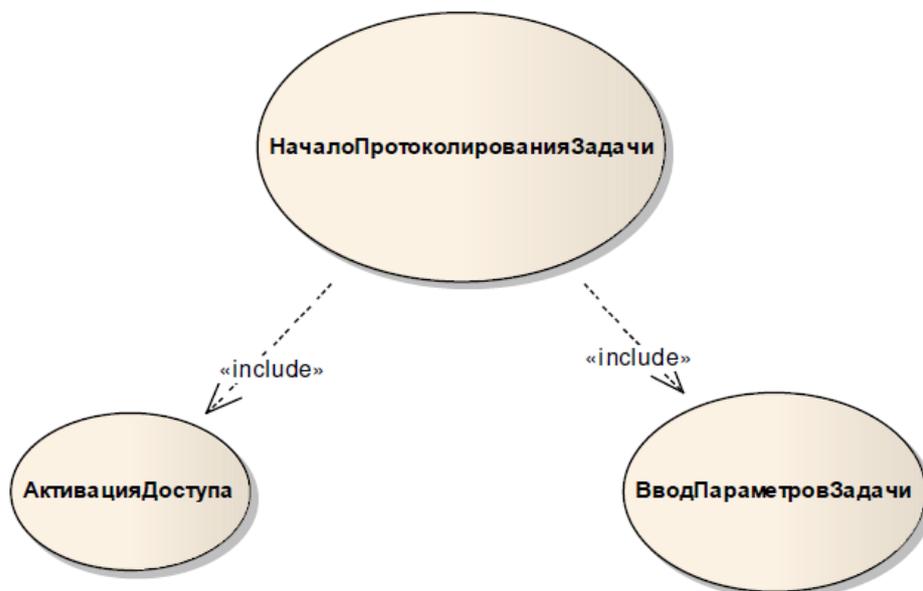


Рис. 3. Отношение композиции

Взаимодействие, описываемое суперэлементом use case, осуществляется при помощи взаимодействий, входящих в подэлемент или подэлементы, причем описание суперэлемента будет ссылаться на все используемые подэлементы. Например, элемент use case под названием «началоПротоколированияЗадачи», созданный для

программы, отслеживающей ход выполнения задач, может использовать элементы «авторизацияДоступа» и «вводПараметровЗадачи». Такой способ моделирования взаимодействий позволяет разделить независимые и почти никак не связанные между собой подзадачи «авторизацииДоступа» и «вводаПараметровЗадачи».

Диаграммы деятельности

Для описания функциональных требований помимо диаграмм вариантов использования используются диаграммы деятельности[4]:

-для описания поведения, включающего большое количество параллельных процессов;

-для анализа варианта использования (описывают последовательность действий и их взаимосвязь);

-для анализа потоков работ(workflow) в различных вариантах использования. Когда варианты использования взаимодействуют друг с другом, диаграммы деятельности являются средством представления и анализа их поведения.

Пример диаграммы деятельности представлен на рис. 4.

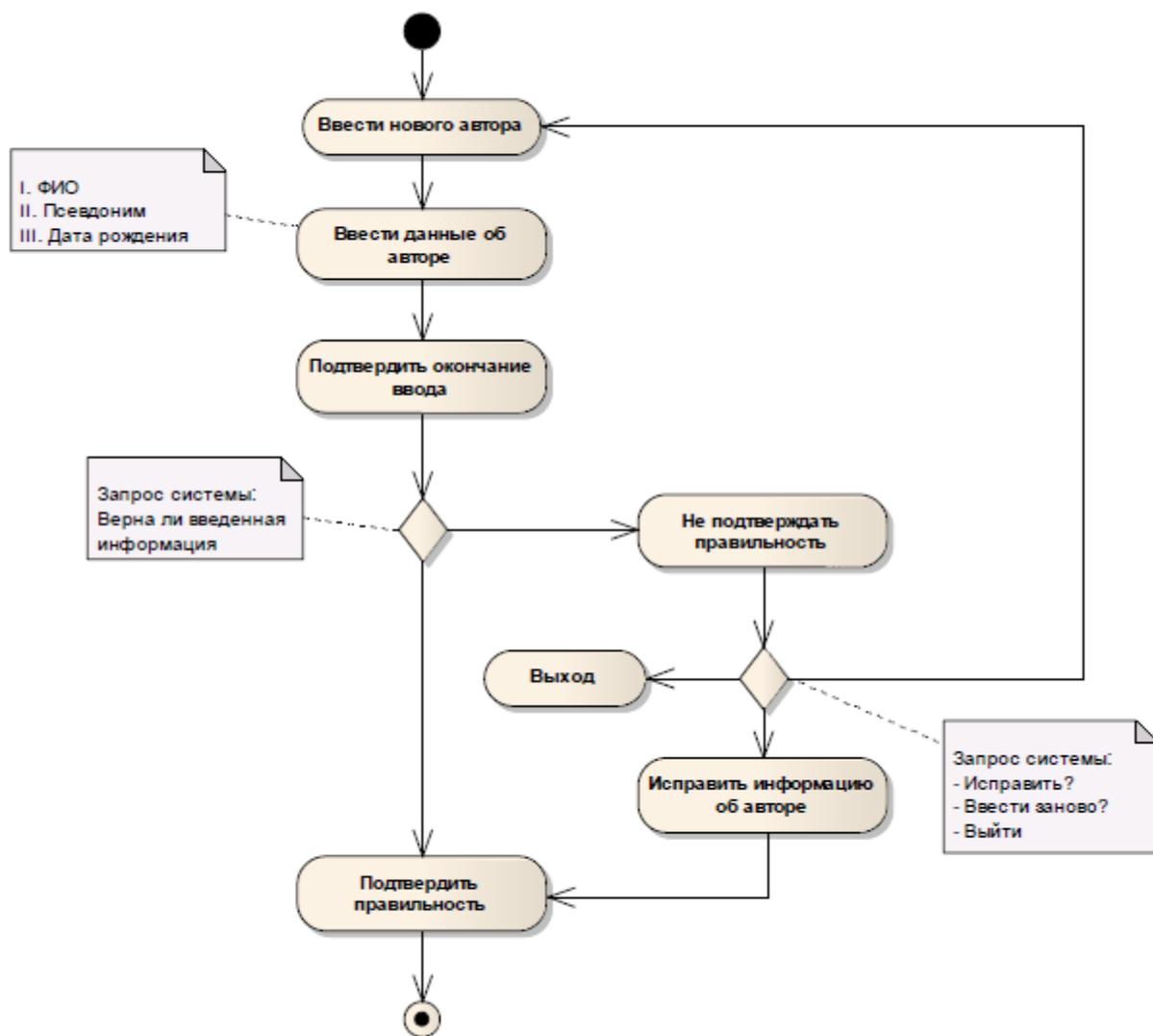


Рис. 4. Диаграмма деятельности

2 .Содержание работы

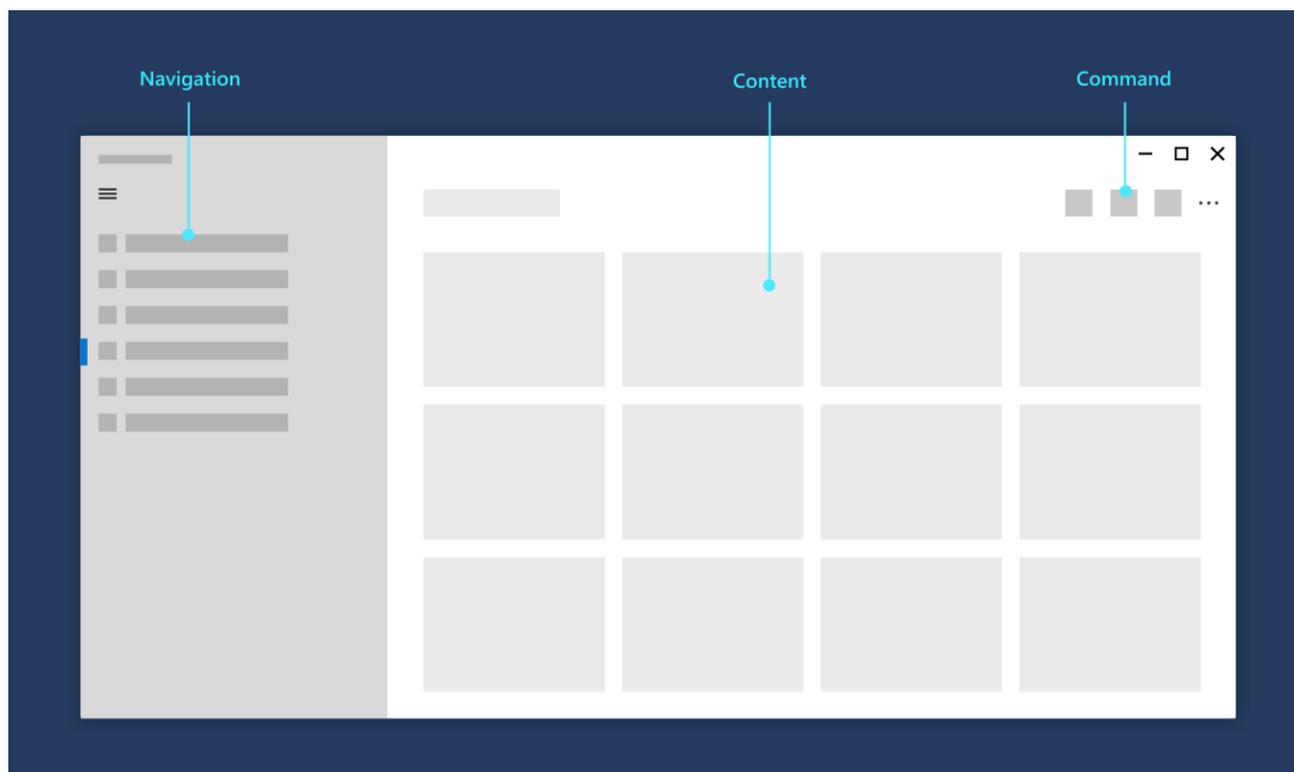
- 1.Изучить основы построения use case диаграмм и диаграмм деятельности.
- 2.Построить use case диаграммы.
3. Построить диаграммы деятельности для варианта использования.
- 4.Оформить отчет о проделанной работе.
- 5.Защитить лабораторную работу.

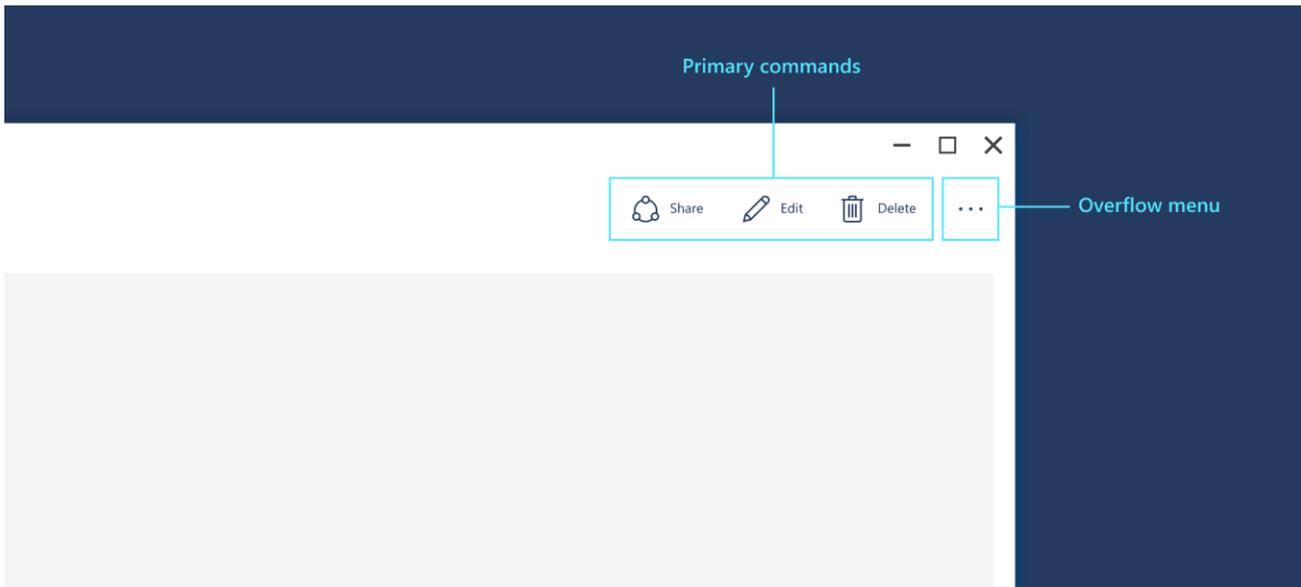
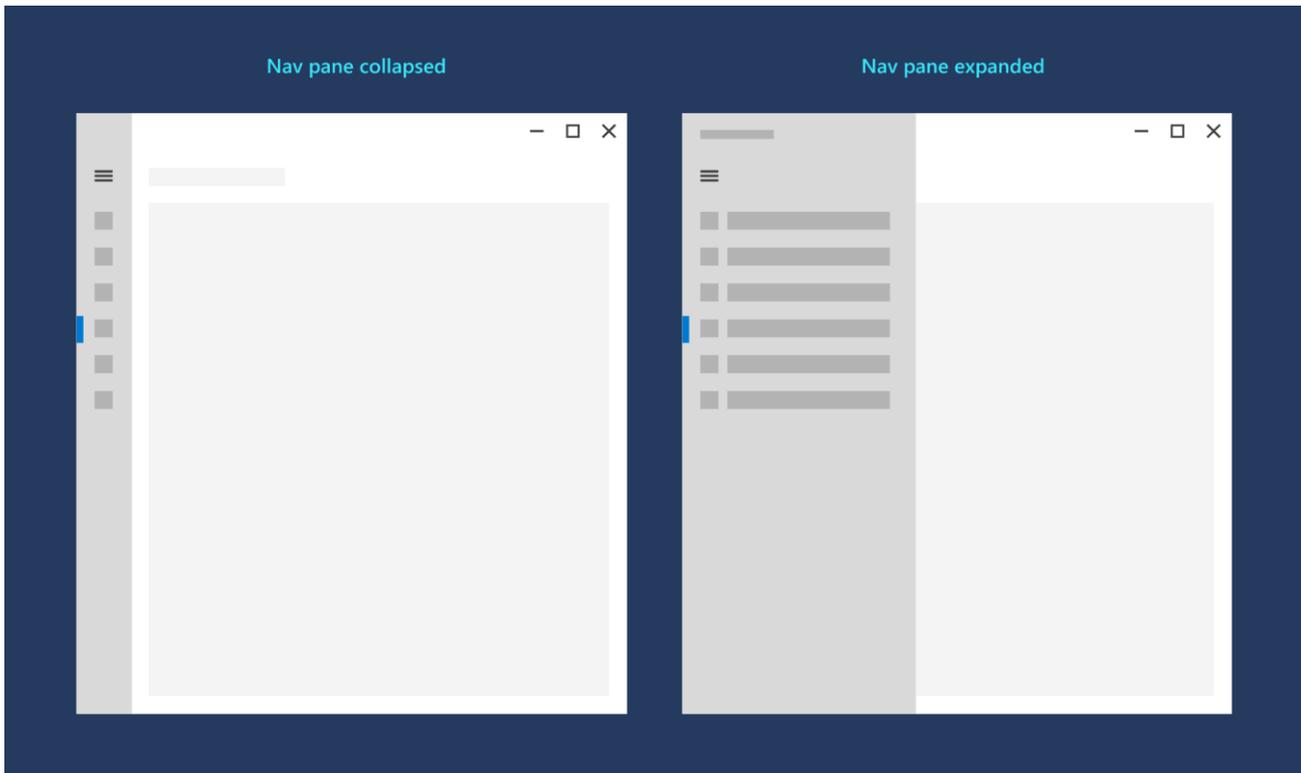
Лабораторная работа № 9 – 10. Разработка макета пользовательского интерфейса.

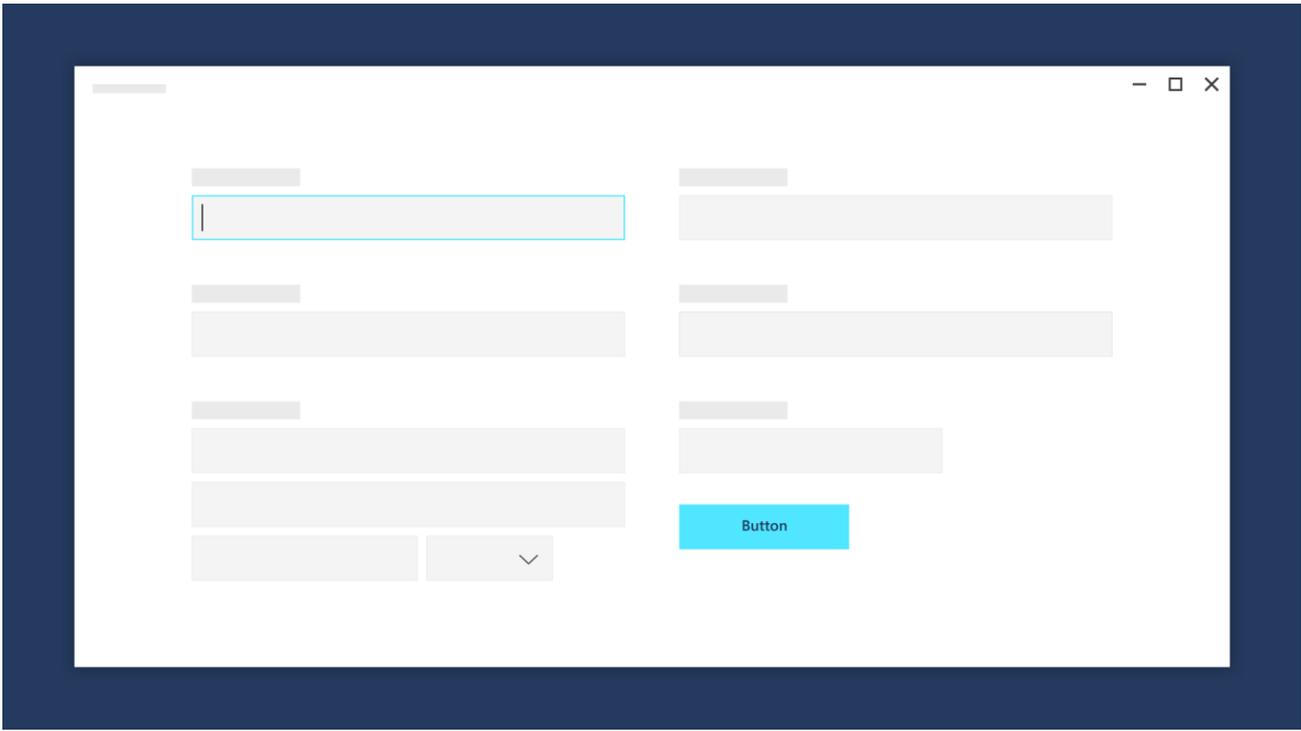
Содержание работы

1. Построить макет всех окон приложений пользовательского интерфейса, в соответствии со своим вариантом, используя графический редактор.
2. Создать описание макета пользовательского интерфейса.
3. Оформить отчет о проделанной работе.
4. Защитить лабораторную работу.

Пример разработки макета:







Лабораторная работа №11 -12. Создание прототипа интерфейса.

1 .Методические указания

При создании интерфейса рекомендуется использовать существующие принципы проектирования пользовательского интерфейса. Далее приводятся шесть принципов, вобравших в себя многое из того, что на данный момент известно о разработке эффективного пользовательского интерфейса. Каждый из них включает в себя несколько связанных между собой идей, более детализированных по сравнению с общими вопросами. Этими общими вопросами являются структура, простота, видимость, обратная связь, толерантность и повторное использование.

Структурный принцип

Организация пользовательского интерфейса должна быть целесообразной, осмысленной и удобной. Она должна базироваться на четких, целостных моделях, очевидных и распознаваемых пользователями. При этом родственные понятия должны быть связаны, а независимые – разделены. Непохожие элементы должны дифференцироваться, а похожие – выглядеть похоже.

Структурный принцип связан с общей архитектурой интерфейса и напрямую отражает представление о пользовательском интерфейсе как о диалоге между разработчиками и пользователями. Организация хороших интерфейсов продумывается очень тщательно, таким образом, чтобы отражать структуру решаемых системой задач и способ мышления пользователей относительно этих задач. Очень часто, особенно при использовании современных визуальных сред разработки, расположение визуальных компонентов внутри форм или диалогов и их распределение между ними оказывается почти случайным и отражает в лучшем случае последовательность, в которой программистами затрагивались те или иные вопросы. По идее, свойства и функции, которые чаще всего используются совместно или рассматриваются пользователями как связанные друг с другом, должны располагаться вместе или, по крайней мере, должны быть четко и ясно взаимосвязаны. Что же до тех элементов, которые в контексте задачи или в сознании пользователя никак не связаны между собой, то они должны быть разнесены в интерфейсе. Подобное должно быть подобно. Похожая информация должна быть организована с помощью похожих решений, а объекты, обладающие похожим поведением, должны иметь общее представление.

Принцип простоты

Следует максимально упрощать управление наиболее распространенными операциями. При этом общение с пользователем должно вестись на понятном для него языке. Должны предоставляться ссылки, логичным образом указывающие на более сложные процедуры.

Процесс проектирования интерфейса – это всегда борьба за компромисс. Упрощение чего-то одного неизбежно приводит к усложнению чего-то другого. Если уменьшить количество меню, увеличится число пунктов в каждом из них. Если сделать маленькими все диалоговые окна, включив в них как можно меньше элементов, любое взаимодействие пользователя с системой обернется для него необходимостью обращаться к большому количеству таких окошек.

Невозможно сделать все на свете простым. Следование принципу простоты

требует от вас знания того, какие задачи выполняются пользователем наиболее часто и какие из них, с точки зрения пользователя, проще. Именно такие задачи следует упрощать, чтобы пользователь мог быстро их решить.

Принцип видимости

Все функции и данные, необходимые для выполнения данной задачи, должны быть видны, чтобы пользователь не отвлекался на дополнительную и избыточную информацию.

Принцип видимости связан с проектированием таких пользовательских интерфейсов, в которых видны все элементы, нужные для выполнения данной задачи. Цель – перейти от философии WYSIWYG (What You See Is What You Get – что видишь на экране, то и получишь в результате) к философии WYSIWYN (What You See Is What You Need – на экране видишь то, что тебе нужно). Интерфейсы WYSIWYN оставляют видимыми те, и только те элементы, которые действительно нужны пользователю для выполнения операции.

С одной стороны, в задачи проектирования входит создание такого интерфейса, на котором были бы явно видны все нужные и важные функции. С другой стороны, хороший интерфейс не должен заваливать пользователя слишком большим количеством возможных вариантов или смущать его избыточной информацией. WYSIWYN-интерфейсы лучше уже тем, что они принимают во внимание ограниченность объема «оперативной памяти» человека и способность узнавать вещи быстрее, чем вспоминать. Нагрузка на долговременную память уменьшается за счет того, что пользователь постоянно видит все необходимые опции и варианты. На кратковременную память нагрузка снижается за счет того, что пользователю не приходится запоминать и затем воспроизводить информацию, содержащуюся в какой-то другой части интерфейса.

Принцип обратной связи

Сообщайте пользователям о действиях системы, ее реакциях, изменениях состояния или ситуации, об ошибках и исключениях, которые важны для них. Сообщения должны быть четкими, краткими, однозначными и написанными на языке, понятном пользователю.

Хорошие пользовательские интерфейсы находятся в диалоге с пользователями, сообщая им о том, что происходит в системе. Принцип обратной связи указывает разработчикам некоторые правила этого диалога.

Практичные системы информируют пользователя о множестве вещей. К примеру, они должны позволять ему узнавать о том, как воспринимаются вводимые им данные. Всякий раз, когда меняется внутреннее состояние системы, и это может оказать какое-либо влияние на работу пользователя, его следует уведомлять об этом, особенно если меняется интерпретация системой его действий. Разумеется, пользователь должен знать о действиях, которые запрещены или игнорируются. При этом принцип обратной связи не может служить оправданием созданию бесконечных окошек сообщений. Информирование пользователя – не самоцель, а способ организации диалога в компактной и естественной форме.

Пользователям также требуются сообщения об ошибках и исключительных ситуациях. Во многих программах эти сообщения, к сожалению, неинформативны и

способны ввести в заблуждение. Можно иногда встретить даже оскорбляющие сообщения, после прочтения которых пользователю может стать не по себе. Вряд ли человек вдохновится, скажем, такой надписью: «Неправильно! Введите корректные данные!». Такое сообщение не только неявно предполагает, что пользователь – какой-то нехороший человек, но и, по сути дела, не дает никакой информации. Здесь не сказано, что именно неправильно и почему.

Грамотно составленные сообщения об ошибках – это еще один пример хорошей организации общения с пользователем. Рекомендации здесь можно дать такие: краткость; язык, понятный пользователю; простота понимания.

Прежде всего информативным должен быть заголовок сообщения. Он должен в сжатой форме описывать проблему, а уже само сообщение должно раскрывать подробности и предлагать способы решения или последовательность корректирующих действий.

Принцип толерантности

Интерфейс должен быть гибким и толерантным. Ущерб, наносимый ошибками пользователя, необходимо снижать за счет возможности отмены и повтора действий и за счет предотвращения появлений этих ошибок путем анализа различных форматов ввода и разумной интерпретации любых разумных действий.

Интерфейс можно делать более или менее толерантным в зависимости от того, какие данные проверяются и когда. Проверка всех полей разом по окончании ввода данных – практика распространенная и иногда оправданная. При этом толерантности системе добавит автоматическая подсветка поля с неправильными данными, установка на него курсора, плюс короткое, информативное сообщение в строке состояния. Больше всего пользователи страдают от программ, которые по окончании ввода во все поля проверяют всю форму, и в случае неправильных данных хотя бы в одном из полей пользователь оказывается снова один на один с пустым бланком. Казалось бы, такое решение ужасно нелепо, но оно встречается очень часто, в том числе и в коммерческих программах.

В целом проверка неиспользуемых полей или полей, которые никак не обрабатываются системой и представляют интерес только для пользователей (в том виде, в каком они были введены), ведет к снижению толерантности ПО. Например, проверка того, что в поле примечаний присутствуют только буквенно-цифровые символы, избыточна. Кроме того, если пользователь вдруг захочет выделить что-нибудь в этом поле спецсимволом или с помощью псевдографики, у него возникнут проблемы.

Принцип повторного использования

Следует многократно использовать внутренние и внешние компоненты и принципы поведения системы, поддерживая устойчивость осмысленно, а не просто за счет избыточности. Это способствует уменьшению объема информации, которую пользователям приходится запоминать и о которой приходится думать каждый раз заново.

Применяя повторное использование внешних и внутренних компонентов и решений, распространяющихся на всю систему, разработчик может создать не только более целостный интерфейс, но и более дешевый продукт. Стремление к одной лишь устойчивости повышает стоимость разработки, да и в ряде случаев оказывается

сизифовым трудом. Нужно стремиться к устойчивости в контексте решаемых системой задач и области ее использования, устойчивость ни в коем случае не может быть самоцелью, иначе она может выглядеть несколько глуповато и даже приводить, как ни странно, к неудачным с точки зрения непротиворечивости системам.

Многие принятые стандарты и общепризнанные компоненты пользовательского интерфейса являют собой примеры неудавшихся попыток реализации устойчивости. Стандартные программные платформы порой навязывают разработчика плохо продуманные и неудачно спроектированные решения. Самые обычные диалоговые окна, скажем, могут обеспечивать целостность и при этом быть весьма низкосортными, выбор стандартных горячих клавиш оказывается случайным и никак не соответствующим принципу устойчивости.

2 .Содержание работы

1 . Разработать прототип интерфейса (в среде Lazarus или web-интерфейс) в соответствии с основными принципами проектирования интерфейса. Интерфейс должен быть достаточен для выполнения всех сценариев.

2 . Оформить отчет о проделанной работе, включающий в себя пример использования каждого из 6 основных принципов проектирования интерфейса.

3 . Защитить лабораторную работу.