



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»
(ДВФУ)**

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Департамент математического и компьютерного моделирования

О Т Ч Е Т

по проектной работе по дисциплине «Программная инженерия»
направление подготовки 09.03.03 «Прикладная информатика»
профиль «Прикладная информатика в компьютерном дизайне»

Выполнили студенты
гр. Б9121-09.03.03пикд

__Соченко А. С._____
(Ф.И.О.) (подпись)

__Безрукова А. Л._____
(Ф.И.О.) (подпись)

__Морокова В. А._____
(Ф.И.О.) (подпись)

Отчет защищен с оценкой: Преподаватель:

(должность, уч.звание)

(Ф.И.О.) (подпись)

«____»_____2023г.

г. Владивосток
2023

Оглавление

1. Введение.....	4
1.1. Глоссарий.....	4
1.2. Описание предметной области.....	4
1.3. Неформальная постановка задачи.....	6
1.4. Обзор существующих методов решения.....	7
1.5. План работ.....	9
2. Функциональные требования к приложению.....	12
3. Спецификация данных.....	13
3.1 Описание формата данных.....	13
3.2. Описание сущности.....	13
3.3. Описание протокола.....	14
4. Требования к окружению.....	15
4.1. Требования к аппаратному обеспечению.....	15
4.2. Требования к программному обеспечению.....	15
4.3. Требования к пользователям.....	15
4.4. Организационные требования.....	16
5. Архитектура системы (Общие требования).....	17
6. Требования к интерфейсу.....	18
7. Прочие требования.....	19
7.1. Требования к надёжности.....	19
7.2. Требования к безопасности.....	19
7.3. Требования к производительности.....	20
8. Проект.....	21
8.1. Средства реализации.....	21
8.2. Структуры данных.....	21
8.3. Модули и алгоритмы.....	21
8.4. Стандарт кодирования.....	22
8.5. Проект интерфейса.....	22
9. Реализация и тестирование.....	23
9.1. Вычислительный эксперимент.....	23
Заключение.....	25
Список литературы.....	26

1. Введение

1.1. Глоссарий

- 1) Категория — понятие, обобщающее места посещения во Владивостоке по определенному признаку.
- 2) Маршрут — путь следования объекта, учитывающий направление движения относительно географических ориентиров или координат, с указанием начальной и конечной точек.
- 3) Подкатегория — более узкая категория, по отношению к материнской, дочерняя категория.
- 4) Сложный маршрут — маршрут, содержащий в себе начальную, конечную и две промежуточные точки.
- 5) Цель путешествия — тематическое исследование туристом местности, культуры, природы, страны, региона в целом.

1.2. Описание предметной области

В Большой Советской энциклопедии [1] можно встретить следующее определение термина путеводитель: путеводитель — краткое справочное издание. Путеводители по стране, городу и т.п., предназначенные главным образом для туристов, содержат сведения о местных достопримечательностях, учреждениях культуры, путях сообщения, отелях и т.п. В современном понятии, путеводитель (гид) — это печатный, электронный или аудиовизуальный справочник о каком-нибудь городе, историческом месте, музее, туристическом маршруте. Главная задача гида — помочь сориентироваться среди наблюдаемого и обозреть все, что следует. Используется, в основном, туристами для лучшего ориентирования в незнакомой местности. Его композиция часто подчинена рекомендуемым маршрутам осмотра.

Электронные путеводители могут выпускаться с конкретными целями:

- пешие экскурсии по городу;
- автомобильные туры;

- музеи, галереи, выставочные залы;
- музеи-заповедники и национальные парки;
- курортные зоны;
- крупные развлекательные центры;
- соревнования по ориентированию;
- масштабные выставки, фестивали и другие мероприятия.

Электронный путеводитель - широкое понятие. Оно может иметь разные формы:

1. Аудиогид - воспроизводится на любом устройстве, поддерживающем mp3. Аудиоэкскурсия - это маршрут, состоящий из остановок, каждой из которых соответствует один трек.
2. Электронный буклет - на экране воспроизводится файл буклета, с картой и текстом (обычно в pdf-формате).
3. Интерактивный справочник - на экране воспроизводится интерфейс программы с доступом к карте и справочной информации.
4. Мультимедийный путеводитель - на карте размещены гипертекстовые ссылки, открывающие фотографии, описание, звуковой комментарий, видеоролик.
5. Мультимедийный навигатор (GPS-путеводитель) - определяет местонахождение экскурсанта, показывает дорогу и автоматически включает звуковой комментарий.

Следует различать так называемые "умные путеводители", которые сами могут определить, где находится человек, и путеводители с ручным управлением.

Под открытым небом используют путеводители, имеющие спутниковую навигацию. Сейчас действует американская система GPS и российская ГЛОНАСС, однако сила сигнала спутниковой навигации очень мала - в помещениях GPS не действует. Задача автоматического ориентирования внутри помещений решается одним из методов альтернативной навигации:

1. Беспроводные сети (Wi-Fi). Принцип работы: в здании устанавливается сервер и беспроводные точки доступа (Wireless Access Point). Программа считывает через точки доступа информацию, хранящуюся в базе данных

сервера. Плюсы: не надо хранить много информации, работает быстрее; базу данных можно обновлять централизованно. Минусы: дорого, сложная технология.

2. Навигация по мобильной связи (GSM). Местоположение определяется по близости к базовой станции сотовой связи.
3. Радиочастотная идентификация (RFID). Навигация ведется по специальным RFID-меткам. Это чипы, которые прикрепляются к объектам. Приемник считывает информацию с этих чипов. Такая система активно используется в логистике, на больших складах, в транспорте.

Электронный путеводитель может быть в виде интернет-сайта или программы. Программа устанавливается на мобильное устройство (КПК, смартфон, iPod, навигатор), персональный компьютер (ноутбук) или запускается с диска. Достоинством является автономность, приспособленность для конкретной цели и устройства. Недостатком - программа обычно платная, данные могут устаревать, не всегда хорошо устанавливается, бывают сбои.

В данный момент не существует приложений-путеводителей, которые могли бы в зависимости от интересов пользователя строить оптимальный маршрут по городу, проходящий через все важные для человека пункты сразу. Появление такого приложения значительно облегчило бы жизнь людям, которые хотят изучить конкретные места города за небольшой промежуток времени, а также позволило бы подбирать лучшие варианты мест той или иной категории в зависимости от отзывов пользователей.

1.3. Неформальная постановка задачи

Требуется разработать проект и реализовать мобильное приложение, которое по интересам пользователя составляет оптимальный маршрут по городу Владивосток. Приложение должно:

- Использовать данные геолокации пользователя;
- Автоматически подбирать места для посещения по интересам пользователя;

- Составлять сложные маршруты при опоре на подключенный картографический сервис Яндекс.Карты;
- Выводить рейтинг мест на основе отзывов пользователей;
- Добавлять понравившиеся маршруты в избранное.

1.4. Обзор существующих методов решения

В процессе работы было найдено (СКОЛЬКО-ТО) мобильных приложений-аналогов. Среди этих приложений можно выделить девять главных конкурентов по функционалу: Foursquare, Sputnik8, Google Maps, Яндекс.Карты, 2ГИС, Culture Trip, Tourist Eye, MAPS.ME и Trip Advisor.

- Foursquare — социальная сеть с функцией геопозиционирования, предназначенная в основном для работы с мобильными устройствами. Если мобильный телефон не оборудован GPS-навигацией, то местоположение определяется с помощью сервиса LBS. Пользователи отмечают в различных заведениях с помощью мобильной версии веб-сайта, SMS-сообщения или же специального приложения, разработанного под определённую ОС мобильного устройства. Каждая такая отметка позволяет пользователю зарабатывать foursquare-баллы, а в некоторых случаях и «бейджи».
- Sputnik8 — это бесплатный сервис для поиска и онлайн бронирования экскурсий, персональных гидов, билетов в музеи и достопримечательности на русском языке. В приложении собрано более 14 тысяч вариантов для досуга в более чем 730 городах мира.
- Google Maps — набор приложений, построенных на основе бесплатного картографического сервиса и технологии, предоставляемых компанией Google. Сервис представляет собой карту и спутниковые снимки планеты Земля. Для многих регионов доступны высокодетализированные аэрофотоснимки (снятые с высоты 250—500 м), для некоторых — с возможностью просмотра под углом 45° с четырёх сторон света.

- Яндекс.Карты — поисково-информационная картографическая служба Яндекса. Есть поиск по карте, информация о пробках, отслеживание городского транспорта, прокладка маршрутов и панорамы улиц крупных и других городов.
- 2ГИС - это подробная карта городов России, включающая в себя сведения о различных организациях (названия, адреса, телефоны, адреса электронной почты и web-сайтов, режим работы), а также карту с возможностью прокладки маршрутов, включая варианты проезда с использованием городского транспорта.
- Culture Trip — помощник для представителей в основном среднего класса. Оно помогает людям определиться с выбором интересных мест по всему земному шару. Авторы статей, публикуемых в приложении, живут по всему миру. К своим публикациям они прилагают живые фото, видео, анимации. Одним словом, делают так, чтобы потребитель заинтересовался тем или иным объектом.
- Tourist Eye — это специальный онлайн сервис для планирования путешествий. С его помощью легко разработать маршрут, выбрать самые интересные места. Каждый пользователь самостоятельно может выбрать тот маршрут, который ему подходит и интересен, заранее спланировать свой отдых.
- MAPS.ME — бесплатное приложение для мобильных устройств на основе свободной географической карты OpenStreetMap. Приложение умеет ориентироваться по GPS, строить маршруты для автомобилей и пешеходов между двумя заданными точками (в том числе, между разными регионами и странами), проводить поиск по объектам, экспортировать и импортировать данные в формате KML, а также предоставлять подробную информацию о заведениях на карте. Доступна голосовая навигация.
- Trip Advisor — сайт путешествий, предоставляющий своим пользователям возможность спланировать поездку в любую страну мира. Услуги сайта бесплатны для пользователей, создающих большую часть контента, также

сайт поддерживается за счёт рекламы. TripAdvisor был одним из первых сайтов, внедривших принцип пользовательского контента.

Все мобильные приложения-аналоги также были разобраны в сравнительной таблице, приведённой ниже.

	Foursquare	Sputnik8	Google Maps	Яндекс.Карты	2ГИС	Culture Trip	TouristEye	MAPS.ME	Trip Advisor	<i>YourPlace</i>
Учёт пожеланий	+	-	+	+	+	+	+	+	+	+
Составление маршрута	-	-	+	+	+	+	+	+	-	+
Больше двух точек в маршруте	-	-	-	+	-	+	+	-	-	+
Подкатегории	-	-	-	-	+	+	-	-	-	+
Рекомендации	+	-	-	+	-	+	-	-	+	+
Наличие экскурсий	+	+	-	-	-	-	-	-	-	+
Офлайн	-	-	-	-	+	-	+	+	-	+
Отсутствие обязательной регистрации	+	-	-	+	+	+	-	+	+	+

Таким образом, все уже существующие приложения с подобным функционалом лишь частично соответствуют необходимым критериям. Следовательно, разработка нового мобильного приложения, сочетающего в себе все функции, описанные в сравнительной таблице, возможна и необходима для удобства пользования.

1.5. План работ

Дата	Соченко	Безрукова	Морокова
09.03.23	Начало введения в отчете	Критерии Подбор цветов Примерная схема приложения	Критерии Подбор цветов Поиск части источников
16.03.23	Доработка введения Составление плана работ	Дизайн приложения Начало вёрстки приложения	Дизайн приложения Сравнительная схема приложений

	Описание функциональных требований Поиск источников	Описание функциональных требований	Описание функциональных требований
23.03.23	Описание спецификации данных Редактирование отчета	Описание спецификации данных Разработка приложения	Разработка дизайна приложения Редактирование отчета
30.03.23	Описание требований к ПО Разработка приложения	Разработка приложения Разработка дизайна приложения	Разработка приложения Разработка дизайна приложения
06.04.23	Описание средств реализации	Проверка на соответствие условиям	Проверка на соответствие условиям
13.04.23	Разработка приложения	Разработка приложения Доработка макета приложения	Разработка приложения
20.04.23	Разработка приложения Написание отчета	Разработка приложения: создание архитектуры приложения	Разработка приложения: составление базы данных
27.04.23	Разработка приложения Написание отчета	Разработка приложения: подключение карт и составление маршрутов	Разработка приложения: подключение базы данных

04.05.23	Поиск способов вывода приложения в маркетплейсы	Поиск способов вывода приложения в маркетплейсы	Поиск способов вывода приложения в маркетплейсы
11.05.23	Создание формы для сбора ответов во время тестирования приложения	Распространение приложения	Распространение приложения
18.05.23	Проведение взаимного тестирования Написание доклада	Проведение взаимного тестирования Создание презентации	Проведение тестирования потенциальным пользователем Написание доклада
25.05.23	Редактирование отчета	Редактирование презентации	Исправление недочетов приложения
01.06.23	Редактирование доклада Репетиция защиты доклада	Репетиция защиты доклада	Репетиция защиты доклада
08.06.23	Сдача проделанной работы комиссии		
15.06.23	Подготовка к защите		
22.06.23	Защита		
29.06.23			

2. Функциональные требования к приложению

Требуется разработать проект и реализовать мобильное приложение, которое по интересам пользователя составляет оптимальный маршрут по городу Владивосток. Приложение должно обладать следующим функционалом:

1. для пользователя — возможность выбрать места для посещения, поставить свою оценку месту и смена маршрута. Для администратора — возможность редактировать пункты посещения, создание и добавление маршрутов, редактирование оценок мест;
2. использование геолокации пользователя;
3. поддержка совместной работы с сервисом Яндекс.Карты;
4. поддержка базы данных с местами посещения, их координатами и рейтингами;
5. собственный уникальный интерфейс.

3. Спецификация данных

3.1 Описание формата данных

Описать назначение формата, указать предполагаемый носитель (например, файл или оперативная память).

Указать, текстовый или двоичный формат. Для текстового формата указать: кодировку символов, правила использования символов перевода строки и возврата каретки (CR/LF), возможно, базовый формат (например, XML). Для двоичного формата указать способ кодировки многобайтных целых чисел (big- или little-endian), формат вещественных чисел (например, IEEE-754 Single Precision).

Подробно описать синтаксис и семантику каждого элемента данных, привести примеры. Если возможно, привести *таблицу полей* или другое структурированное изображение формата. Описать возможные ошибки в представлении данных и реакцию на них.

3.2. Описание сущности

Термин сущность (таблица) обычно используется при разработке баз данных и объектно-ориентированных систем.

Описать назначение сущности («предназначена для хранения ...» либо «реализует понятие ...»).

Перечислить атрибуты сущности, имеющие смысл в предметной области. При этом не указываются служебные поля БД, невидимые для пользователя (например, искусственный первичный ключ, время последней модификации или номер реплики). Для каждого атрибута указать:

- полное название по-русски (а не имя соответствующего поля БД),
- тип (без детализации, например, не «VARCHAR(200)», а «Строка»),
- ограничения на допустимые значения,
- обязательность,
- доступность для редактирования,
- уникальность (для составных ключей указать «комбинация значений атрибутов А, В, ... должна быть уникальной»),
- значение по умолчанию (если есть).

Если атрибут используется для создания отношения «один ко многим», следует указать его тип как «ссылка на *Название сущности*», а не «Целое число».

Оформить список атрибутов в виде таблицы, содержащей все или часть из перечисленных характеристик. Часто повторяющиеся значения вынести в текст после таблицы. Например «Значения атрибута *Фамилия* должно быть уникальным» или «Для редактирования доступны все атрибуты, кроме атрибута *Дата создания объекта*».

Описать дополнительные требования к данным сущности. Перечислить объекты сущности, автоматически создаваемые при установке системы, и значения их атрибутов.

Привести *диаграмму классов* (или объектов).

3.3. Описание протокола

Описать назначение протокола.

Указать стороны, общающиеся при помощи данного протокола, а также базовый протокол более низкого уровня (например, согласно модели OSI [2]). Формализовать отображение описываемого протокола на базовый. Описать формат сообщений протокола и общие правила их интерпретации, аналогично п. 4.1. Указать динамические требования к протоколу (задержки, тайм-ауты, повтор сообщений и т. д.). Указать наличие хранимого состояния в протоколе (stateful или stateless).

Перечислить список всех сообщений протокола, для каждого сообщения указать содержащиеся в нём значения фиксированных полей (например, тип пакета) и семантику переменных полей, а также сопутствующие изменения в состояниях сторон. Если формат сообщений достаточно сложен, описать его аналогично структуре данных.

Обратить внимание на возможные ошибки и непредвиденные ситуации, в частности, непредвиденное прекращение функционирования одной из сторон и разрыв канала связи.

Привести, по меньшей мере, одну *динамическую диаграмму* (например, *диаграмму состояний*).

4. Требования к окружению

Раздел требований к окружению содержит, в отличие от последующих разделов, не требования к системе, а, напротив, требования, предъявляемые системой к окружающему миру.

4.1. Требования к аппаратному обеспечению

Описать *минимально* *необходимую* и, возможно, рекомендуемую аппаратную конфигурацию для *эффективной* работы описываемой системы (т. е. такую конфигурацию, в которой система не только запускается, но и способна производить полезные действия). Особенно выделить нестандартное оборудование, необходимое для работы, например, звуковую карту или интерфейс специализированного измерительного прибора. Если нестандартные устройства необходимы для выполнения лишь некоторых функций системы, то указать эти устройства и функции отдельно. Если система требует более одного аппаратного узла (например, клиент-серверная или кластерная архитектура), описать требования к каждому узлу по отдельности, а также требования к каналам связи между узлами.

4.2. Требования к программному обеспечению

Описать базовое программное обеспечение, установка которого необходима для функционирования описываемой системы, например, операционную систему, веб-браузер, СУБД и т. п. Для каждого компонента ПО указать аппаратный узел из п. 2.1, на котором он будет выполняться. Привести *точные версии* компонентов, на которых производилось тестирование, использовать фразу «и выше» для указания на то, что описываемая система будет, вероятно, работать и с более поздними версиями.

В случае разработки библиотеки требования к программному окружению включают компилятор или среду программирования, для которой предназначена библиотека.

Если система переносима между различными платформами, то указать, во-первых, полные и точные программные конфигурации платформ, на которых производилось тестирование, и, во-вторых, базовый компонент, от переносимости которого зависит переносимость описываемой системы. (Например, «и другие ОС, поддерживающие стандарт POSIX» или «и другие платформы, на которых работает PHP 4»).

4.3. Требования к пользователям

Описать категории пользователей системы, при этом указать для каждой категории:

- название категории (используется также термин «роль»);
- краткое обозначение исполняемых функций как в системе, так, где это применимо, и в организации (например «ввод первичных данных»);
- доступные возможности системы;
- требования, предъявляемые к знаниям, умениям и навыкам данного пользователя.

Не указывать тривиальные навыки (например, умение читать) там, где это очевидно. Напротив, специально указать возможное отсутствие какого-либо базового навыка у пользователей (например, «программа предназначена для детей дошкольного возраста, не умеющих читать», «программа поддерживает работу пользователей с дефектами зрения»). Обратить особое внимание на знание пользователями языка и терминологии, используемой в интерфейсе.

4.4. Организационные требования

Указать изменения в структуре и производственном процессе организации-заказчика, необходимые для успешного внедрения системы. Например, «требуется введение ставки системного администратора с приблизительной занятостью в половину рабочего дня» или «ежедневное резервное копирование данных необходимо вменить в обязанность главному менеджеру». В случае отсутствия или незначительности таких изменений раздел опустить.

5. Архитектура системы (Общие требования)

Описать разделение системы на видимые с точки зрения пользователя подсистемы (например, клиентскую и серверную часть), кратко указать функции каждой подсистемы и перечислить общие для всех подсистем требования. Привести *диаграмму пакетов* и/или *диаграмму развёртывания*. Если система создаётся группой разработчиков, указать общие требования либо сослаться на документ, описывающий их, а также указать подсистемы, разрабатываемые автором отчёта. Все изложенные в последующих разделах требования должны относиться только к этим подсистемам.

Данный раздел можно опустить при описании простой системы, не подразделяющейся на более мелкие подсистемы.

6. Требования к интерфейсу

Интерфейс приложения должен быть выдержан в специально подобранной спокойной цветовой гамме, а также соответствовать «морской» тематике. Все функции интерфейса должны быть доступны при использовании для ввода с экрана смартфона. Приложение должно обладать следующим интерфейсом:

1. две кнопки на главном экране: «Места города» и «Построить маршрут» — пользователь может выбрать желаемое дальнейшее действие;
2. кнопка «Места города» — пользователь переходит на экран категорий пожеланий;
3. стрелка «Назад» — пользователь возвращается на предыдущий экран;
4. раздел «Поесть» — пользователь переходит на экран списка точек питания, отсортированных по типу кухни;
5. раздел «Прогуляться» — пользователь переходит на экран списка мест для прогулок;
6. раздел «Заняться спортом» — пользователь переходит на экран списка точек спортивных комплексов;
7. раздел «Развлечься» — пользователь переходит на экран списка точек развлекательных мест;
8. раздел «Культурно отдохнуть» — пользователь переходит на экран списка мест культурного характера;
9. раздел «Посетить экскурсии» — пользователь переходит на экран списка доступных экскурсий по городу Владивосток;
10. кнопка «Построить маршрут» — пользователь переходит на экран выбора трёх точек маршрута;

7. Прочие требования

Указать требования к нефункциональным характеристикам системы, отличающиеся от общепринятых либо имеющие критическое значения для успешной реализации проекта. Любой из подразделов, а также и весь раздел можно опустить в случае отсутствия соответствующих требований.

7.1. Требования к надёжности

Оценить вероятность возникновения, возможные последствия и убытки от нештатных ситуаций (рисков), могущих возникнуть в связи с использованием системы, в том числе:

- отказ оборудования;
- обнаружение ошибок разной степени тяжести;
- утеря или повреждение данных;
- физическое нарушение безопасности;
- резкое изменение экономических условий или производственного процесса.

Сделать вывод о наличии или отсутствии необходимости мероприятий по управлению данными рисками. Перечислить и описать список мероприятий, для каждого мероприятия указать: процедуру проведения, предполагаемую периодичность, категории пользователей, ответственных за его выполнение. Например: «ежедневное резервное копирование базы данных на магнитную ленту производится в конце рабочего дня Оператором» или «для обеспечения физической безопасности сервера он должен быть помещён в охраняемое помещение, доступ к которому имеет только Администратор и Директор».

Отдельно указать требования к устойчивости программного продукта к различным нештатным ситуациям, действиям по обработке ошибок и восстановлению после сбоев. Например, «неожиданное отключение питания должно приводить к потере только данных только из тех записей БД, которые редактируются в этот момент». Если значительная часть требований к надёжности выполняется за счёт базового программного обеспечения, например, СУБД или ОС, указать это.

7.2. Требования к безопасности

Если система предназначена для хранения и обработки ценных данных либо для управления критическими процессами, рассмотреть отдельно риски, связанные со злоумышленным несанкционированным доступом к системе, в том числе:

- доступ на чтение к конфиденциальным данным (например, финансовые, медицинские записи);
- доступ на запись к открытым, но важным данным (например, данные об успеваемости, результаты выборов);
- перехват управления критическими процессами;
- разрушение данных и вандализм;
- временное нарушение работы системы без разрушения данных (Denial-of-Service).

Указать возможные векторы атаки и необходимые для предотвращения атак изменения функциональности (например, шифрование передаваемых или хранимых данных, проверка паролей пользователя на сложность, перенос административных функций из Web-интерфейса в локальное приложение и т. д.).

7.3. Требования к производительности

Определить диапазоны количественных параметров размеров задач, для решения которых предназначена система, в том числе объём базы данных, количество одновременно работающих пользователей, скорость поступления внешних сигналов и т. п.

В предположении, что система будет использовать рекомендуемую конфигурацию оборудования из раздела 2.1, указать минимальные требования к количественным показателям эффективности системы. В число показателей могут входить: время отклика, скорость выполнения транзакций, время обработки запроса, объём используемой оперативной памяти и дискового пространства и др. Для каждого из показателей может быть указано предельно допустимое и ожидаемое среднее значение. Например, «среднее время отклика системы на действие пользователя должно быть 2-3 секунды, максимальное — не более 5 секунд». Если часть показателей зависит от внешних факторов, (например, качества TCP/IP соединения), отметить это и привести требования для случая их оптимального сочетания. Дополнительно возможно указание ожидаемых реальных показателей в различных условиях.

8. Проект

Раздел содержит данные о внутренней структуре программной системы и принятых при разработке решениях, не проистекающих непосредственно из требований. В отличие от требований, предложения в разделе «Проект» должны быть сформулированы не в побудительной, а в утвердительной форме, (например, вместо фразы «система должна состоять из 5 модулей» следует написать «система состоит из 5 модулей»). Некоторые подразделы могут отсутствовать, если реализация соответствующих требований тривиальна или непосредственно определяется самими требованиями.

Привести во всех подразделах все необходимые *статические и динамические диаграммы* (например, *диаграмму компонентов*).

8.1. Средства реализации

Перечислить и сравнить языки программирования, среды разработки, библиотеки программ, СУБД и т.п., рассматривавшихся в качестве возможных средств реализации системы. Учесть характеристики открытости и цены средства, скорости и простоты разработки, надёжности и эффективности конечного программного продукта. Если проводилось сравнительное тестирование или апробация средств, привести результаты. Документировать и обосновать осуществлённый выбор средств.

8.2. Структуры данных

Описать внутренние структуры данных в порядке, аналогичном спецификации данных. Описать, если необходимо, низкоуровневую структуру базы данных (в терминах таблиц и полей вместо сущностей и атрибутов). Привести точные и детализированные типы данных и описания форматов.

Описать нетривиальные структуры данных, используемые алгоритмами программной системы. Аргументировать их выбор.

8.3. Модули и алгоритмы

Описать общую структуру системы или каждой из подсистем. Перечислить, в зависимости от размера и сложности системы, все модули и классы и/или процедуры и функции с кратким описанием назначения каждого пункта. Описать внутренние интерфейсы между частями системы.

Если при разработке были применены или созданы нетривиальные алгоритмы, описать их. Для известных алгоритмов привести ссылки, для новых — сравнение с существующими и анализ их свойств, эффективности, преимуществ и недостатков.

8.4. Стандарт кодирования

Кратко описать используемый стандарт кодирования, указать его источник (например, принят в организации-заказчике, рекомендуется разработчиками используемого языка программирования), привести ссылку на литературу. Обратить внимание на уникальные для данного проекта элементы стандарта, особенно соглашения об именовании.

Если для оформления кода использовались автоматические инструменты (например, среды программирования с автоматической генерацией кода, программы автоматического форматирования кода), указать эти инструменты и параметры их вызова.

8.5. Проект интерфейса

Перечислить и обосновать дизайнерские решения, принятые в процессе разработки интерфейса. В том числе цветовую гамму, шрифты, размеры, форму и расположение элементов управления, и т. д. Привести, по меньшей мере, по одному *снимку экрана* (*screenshot*) для каждого существенно отличающегося модуля или режима работы. При получении снимков экрана следует:

- следить за соответствием версии программы, изображаемой на снимке, с версией, описываемой в отчёте;
- включать в снимок только окно программы, а не весь экран операционной системы (в частности, следует удалять изображение Windows taskbar и меню/панели инструментов браузера);
- уменьшать окно программы, особенно по вертикали, таким образом, чтобы оно не содержало пустого пространства;
- по возможности подбирать цветовую схему, хорошо отображающуюся на черно-белую печать (в частности, желательно избегать крупных участков с чёрным фоном, а также низкоконтрастного текста на светлом фоне).

9. Реализация и тестирование

Привести данные о физических характеристиках текущей версии системы:

- объём *написанного автором* кода в килобайтах и строках, отдельно по каждому языку программирования,
- для работ, заключающихся в изменении существующей системы, объём *изменённого автором* кода в строках,
- объём *автоматически сгенерированного* кода в килобайтах и строках, отдельно по каждому языку программирования (сюда входят, например, файлы *.dfm в системе Delphi),
- количество модулей, форм, экранов, страниц сайта и т. п.,
- количество автоматических тестов,
- количество и объём, в килобайтах, программных компонент,
- фактическое быстроедействие и затраты оперативной памяти, на нескольких примерах, сравнить с требованиями п. 7.2.
- фактическое количество пользователей, кроме разработчиков системы.

Если данная система разрабатывалась как часть более крупной, привести количественную (например, «около 15% кода») и качественную оценку вклада данной подсистемы.

Указать методику тестирования: по белому или чёрному ящику, бета-тестирование, случайное тестирование. Описать процедуру тестирования (вручную или автоматически), его объём и результаты.

Если проводилось внедрение, описать процесс внедрения, полученный технико-экономический эффект, отзывы пользователей. Сослаться на *Акт о внедрении*, который должен прилагаться к работе.

Сделать вывод об успешности реализации программной системы.

Если сложность описываемой системы невелика, данный раздел можно опустить, и перенести данные о характеристиках системы и её внедрении в «Заключение».

9.1. Вычислительный эксперимент

Данный раздел наиболее характерен для научно-исследовательских работ и часто является центральной частью таких работ.

Следует указать цели эксперимента, экспериментальную гипотезу (если есть). Типичными целями являются: демонстрация возможности (приблизительного) решения поставленной задачи разработанным алгоритмом, подбор оптимальных

параметров алгоритма, оценка производительности и оптимизация, сравнение различных вариантов алгоритма, оценка качества прогнозирования, выполняемого алгоритмом.

Описать методику проведения эксперимента, обратить особое внимание на возможность её воспроизведения независимыми исследователями. При исследовании производительности — подробно описать использованную программно-аппаратную конфигурацию и методику измерений.

Привести результаты эксперимента в виде набора таблиц и графиков. При построении графиков, особенно сравнительных, обратить внимание на корректный выбор масштаба по осям.

Проанализировать результаты, сделать выводы о достижении целей эксперимента. Если целью эксперимента был подбор оптимальных параметров или вариантов алгоритма, перечислить полученные рекомендации.

Заключение

Раздел не нумеруется. Начать раздел фразой «Таким образом, в процессе *курсовой / дипломной / др. работы* мною было ...», за которой перечислить виды деятельности, выполненные в рамках работы. Отделить в списке учебную деятельность («изучено», «углублены знания / повышены навыки в области...») от производственной («разработано», «спроектировано», «реализовано», и т.п.)

В перечислении избегать общих выражений («изучена предметная область»), а вместо этого использовать конкретные («изучены основы банковского дела и схема работы на примере банка Х»).

Кратко перечислить основные характеристики и достоинства разработанной системы, привести данные о её внедрении и достигнутом за счёт него эффекте, указать пути дальнейшего развития системы. В случае отсутствия п. 8 «Реализация», привести краткие сведения об объёме и сложности системы.

Список литературы

1. [https://ru.wikipedia.org/wiki/Большая советская энциклопедия](https://ru.wikipedia.org/wiki/Большая_советская_энциклопедия)
- 2.