#### Задание **6**

#### Задание выполняется в группе из 3-4 человек, раздел 5 выполняется совместно, остальные разделы распределяются между участниками группы.

Написать документ-концепцию

Документ-концепция описывает приложение в общих чертах, а также содержит описания целевых рынков, пользователей системы и функций приложения (как предлагаемых к реализации в версии 1, так и выявленные), ограничения системы.

##### **Содержание документа-концепции**

1. Введение  
   В данном разделе необходимо представить общую характеристику документа-концепции в целом  
   1.1. Цель документа-концепции  
   Цель данного документа состоит в сборе, анализе и определении высокоуровневых потребностей пользователей и функций продукта.  
   1.2. Общая характеристика продукта  
   В данном разделе определяется цель приложения, его версия и новые предоставляемые функции. Здесь следует

* указать продукт или приложение, которое создается или изменяется;
* дать общее описание того, что продукт будет делать и, если необходимо, чего не будет делать;
* описать применение продукта, в том числе достижимые с его помощью выгоды, цели и задачи.

1.3. Ссылки  
Этот подраздел содержит cписок всех документов, упоминаемых где-либо в документе-концепции и список источников, к которым можно обратиться за справками.

1. Описание пользователя  
   2.1. Характеристика рынка/пользователя  
   Здесь необходимо кратко перечислить основные характеристики рынка, которые послужили мотивацией решений, касающихся продукта: описать и указать целевые сегменты, а также оценить объем и перспективы роста рынка, ориентируясь на число потенциальных пользователей или количество денег, которые в настоящее время тратят ваши заказчики при решении задач.

2.2. Описания пользователей  
Здесь следует описать все типы пользователей. Для каждого типа пользователей указать

* Технический уровень и опыт
* Основные обязанности
* Тенденции, упрощающие или усложняющие работу пользователя
* В чем пользователь видит успех и как пользователь вознаграждается

2.3. Среда пользователя

* Сколько человек участвует в выполнении данной задачи? Изменится ли их число?
* Сколько времени длится цикл выполнения задачи? Изменится ли это?
* Существуют ли некие уникальные ограничения среды: на мобильную связь, по работе вне помещения, в полете и т.д.
* Какие системные платформы используются в настоящее время? Какие платформы предполагается использовать в будущем?
* Какие еще приложения используются? Должно ли ваше приложение объединяться с ними?

2.4. Основные потребности пользователя  
Следует перечислить основные проблемы или потребности так, как они осознаются пользователем. Для каждой проблемы нужно прояснить следующие моменты

* В чем причины данной проблемы?
* Как она решается в настоящее время?
* Какие решения представляет себе пользователь?

2.5. Альтернативы и конкуренты  
Нужно указать возможные альтернативы поведения пользователя. Среди них может быть покупка продукта конкурентов, создание собственного решения или просто сохранение существующей ситуации. Опишите основные преимущества и недостатки каждого варианта с точки зрения конечного пользователя.  
2.5.1. Конкурент 1

1. Характеристика продукта  
   В данном разделе предлагается общее описание возможностей продукта, интерфейсов с другими приложениями и конфигураций систем.  
   3.1. Общее описание продукта  
   В данном подразделе следует описать, как продукт взаимодействует с другими связанными с ним продуктами и средой пользователя. Если продукт является независимым и самодостаточным, это необходимо указать. Если продукт является компонентом более крупной системы, в данном подразделе необходимо описать, как эти системы взаимодействуют, а также указать соответствующие интерфейсы между системами.  
   3.2. Определение позиции продукта  
   Предлагается общее определение, характеризующее на самом высоком уровне абстракции назначение продукта и важность проекта.  
   3.3. Краткий обзор возможностей  
   Краткая характеристика основных возможностей и функций продукта с точки зрения преимуществ.  
   3.4. Предположения и зависимости  
   Описываются предположения, изменение которых приведет к изменению концепции продукта.  
   3.5. Вопросы затрат и цены
2. Атрибуты функций  
   Функции имеют атрибуты, предоставляющие дополнительную информацию, которую можно использовать для оценки, отслеживания и определения очередности предлагаемых для реализации элементов разработки, а также управления ими. Нужно описывать в данном разделе только те атрибуты (и их значения), которые вы выберете, чтобы все участники могли лучше понять содержание каждой функции.  
   4.1. Статус (предложена, принята, включена) (более подробно в файле Атрибуты.docx) 4.2. Приоритет (критический, важный, полезный)  
   4.3. Уровень трудозатрат  
   4.4. Риск  
   4.5. Стабильность  
   4.6. Целевая версия  
   4.7. Кому предназначена  
   4.8. Обоснование
3. Функции продукта  
   Поскольку документ-концепция изучается широким кругом причастных к проекту лиц и служит основой для достижения соглашения, функции должны описываться на естественном языке пользователя. Описание функции должно быть кратким и ясным, как правило, одно-два предложения. Для эффективного управления сложностью приложения мы рекомендуем, чтобы описание возможностей любой новой системы (или усовершенствования существующей) производилось на достаточно высоком уровне абстракции и состояло из 25-99 функций. Эти функции составляют основу для определения продукта, а также управления масштабом и проектом в целом. Каждая из них будет описана более подробно в последующих спецификациях.  
   5.1. Функция 1  
   5.2. Функция 2
4. Основные варианты использования  
   Следует описать несколько основных вариантов использования, которые важны для архитектуры или лучше всего помогут читателю понять, как предполагается использовать систему
5. Другие требования к продукту (ограничения)  
   7.1. Применяемые стандарты  
   7.2. Системные требования  
   7.3. Лицензирование и инсталляция  
   7.4. Требования производительности
6. Требования к документации  
   В данном разделе описывается, какую документацию необходимо разработать для поддержки успешного внедрения приложения.  
   8.1. Руководство пользователя  
   8.2. Интерактивная подсказка  
   8.3. Руководства по инсталляции, конфигурация и файл ReadMe  
   8.4. Маркировка и упаковка
7. Глоссарий  
   Глоссарий описывает все присущие данному проекту термины, в том числе все аббревиатуры, которые могут быть непонятны пользователю или другим читателям данного документа.

Критерии оценки

* Структура документа-концепции соответствует шаблону - 2 балла
* Достаточное описание пользователя и его потребностей (пп 2.2,2.3,2.4) - 3 балла
* Рассмотрено не менее 2 альтернатив - 2 балла
* В п.3 указано достаточное обоснование важности проекта - 1 балл
* Указано не менее 5 функций - 2 балла

Итого 10 баллов максимум

**4. Атрибуты функций**

Как и требования, функции имеют атрибуты, предоставляющие дополнительную ин­формацию, которую можно использовать для оценки, отслеживания и определения очередности предлагаемых для реализации элементов разработки, а также управле­ния ими. Ниже мы описали атрибуты, которые можно использовать в документе-концепции. *Вам нужно описывать в данном разделе* ***только те атрибуты (и их значе­ния)****, которые вы выберете, чтобы все участники могли лучше понять содержание каждой функции.*

**4.1.Статус**

Задается в результате переговоров и рассмотрения руководством проекта. Ин­формация о статусе отражает ход процесса определения базового уровня про­екта. Атрибут статуса функции может иметь следующие значения.

**Предложена.** Используется для описания обсуждаемых функций, которые еще не рассмотрены и не приняты "официальным органом" — рабочей группой, состоящей из представителей команды проекта\* руководства и пользователей или заказчиков.

**Принята.** Возможности, которые "официальный орган" признал полезны­ми и достижимыми и принял к реализации.

**Включена.** Функции, включенные в базовый уровень на данный момент времени.

**4.2.Приоритет**

Приоритеты функций продукта задаются представителями маркетинга, ме­неджером продукта или аналитиком базового уровня. Упорядочение функций по их относительной важности для конечного потребителя открывает диалог между заказчиками, аналитиками и членами команды разработчиков. Приори­теты используются для управления масштабом и определения очередности разработки. Ниже предложена одна из возможных схем задания приоритетов.

**Критический.** Основные функции. Если их не удастся реализовать, сис­тема не будет удовлетворять потребности заказчика. В версии должны быть реализованы все критические функции, в противном случае график являет­ся нереальным.

**Важный.** Функции, важные для успешной и эффективной работы системы в большинстве приложений. Данные функциональные возможности нельзя легко обеспечить иным способом. Если важные функции не войдут в реали­зацию, это может повлиять на удовлетворение пользователя или заказчика результатом работы или даже на доходы от продаж, но выпуск версии не должен задерживаться из-за нехватки некой важной функции.

**Полезный.** Функции, которые нужны в менее распространенных прило­жениях, будут использоваться не так часто или их можно достаточно эф­фективно Заменить другими действиями. Если они не войдут в реализацию,это не окажет заметного воздействия на отношение заказчика или доходы.

***4.3.* Уровень трудозатрат**

Определяется командой разработчиков и используется для управления мас­штабом и определения очередности разработки. Поскольку некоторые функ­ции требуют больше времени и ресурсов, чем другие, оценка количества ко­мандо- или человеко-недель, строк кода или функциональных единиц помогает соразмерить сложность и оценить, что можно, а что нельзя осуществить за оп­ределенный период времени.

**4.4. Риск**

Задается командой разработчиков на основе вероятности того, что данная функция вызовет нежелательные последствия для проекта, такие как превышение средств, отставание от графика или даже закрытие проекта. Большинство менеджеров продукта считают достаточным деление рисков на категории *низкий, средний, высокий,* хотя возможна и более тонкая градация. Иногда риск можно оценить, измеряя меру неопределенности (диапазон) оценок времени работы команды.

**4.5. Стабильность**

Определяется аналитиком и командой разработчиков, исходя из вероятности того, что может измениться данная функция или понимание командой этой функции. Эта информация используется для того, чтобы помочь при опреде­лении приоритетов разработки и выявить те элементы, для которых следую­щим действием должно стать дополнительное исследование.

**4.6. Целевая версия**

Записывается, в какой версии продукта предполагается впервые реализовать данную функцию. Это поле можно использовать, чтобы поместить функции в базовый уровень конкретной версии. Комбинируя этот атрибут с полем статуса, команда может предлагать, записывать и обсуждать для версии различные функции, не приступая к их разработке. Будут реализовываться только функции:, имеющие статус "Включенная", для которых определена целевая версия. При необходимости сокращения масштаба номер целевой версии может быть увеличен, tax что элемент остается в документе-концепции, но его реализация будет отложена на более поздний срок.

**4.7. Кому предназначена**

Во многих проектах функции будут предназначаться "функциональным группам", ответственным за их дальнейшее исследование, написание программных требова­ний, а также, возможно, реализацию. Это помогает членам команды разработчиков лучше понять свои обязанности.

**4.8. Обоснование**

Данное текстовое поле используется для отслеживания источника запраши­ваемой функции. В этом поле записывается объяснение причины существования данной функции или ссылка на него. Например, ссылка может указывать на страницу, номер строки спецификации требований к продукту или временной маркер на видеозаписи важного интервью с клиентом.

**Информационная система  
«Электронный архив»**

**Документ-концепция**

**© Е. П. Моргунов**

История исправлений и дополнений

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Версия | Описание | Автор |
| 01.04.2005 | 0.5 | Исходная версия | Е. П. Моргунов |
| 20.06.2009 | 1.0 | Проведены реструктуризация и дополнение документа с учетом рекомендаций, предложенных в книгах И. Соммервилла и Д. Леффингуэлла (их библиографические описания см. ниже) | Е. П. Моргунов |
|  |  |  |  |

Примечание. Дата выпуска версии 0.5 – приблизительная.

**1. Введение**

**1.1. Цель документа-концепции**

Цель данного документа состоит в сборе и анализе исходной информации для разработки, определении высокоуровневых потребностей пользователей и формулировании функций продукта.

**1.2. Назначение и общая характеристика продукта**

Настоящий продукт предназначен для решения разнообразных задач управления информационным массивом, который накапливается и используется любым человеком в процессе интеллектуальной деятельности (в том числе, и связанной с отдыхом). Информация, интересующая конкретного человека, может содержаться в печатных и электронных изданиях, аудио- и видеоматериалах, рукописях, ксерокопиях и т. д. Зачастую требуется не только отыскать нужную информацию, например, цитату, но также и указать ее источник. Следовательно, возникает необходимость накапливания и структуризации как содержательной информации, ради получения которой и проводится проработка различных источников (их чтение, просмотр, выписывание цитат и т. п.), так и библиографической информации. Ее можно условно назвать метаинформацией.

Программный продукт в технологическом плане будет представлять собой информационную систему, построенную на основе «большой» СУБД и Web-технологий. Это позволит в качестве клиентского места использовать Web-браузер, а при необходимости – организовать удаленный доступ к базе данных через Internet/Intranet. В содержательном плане программный продукт должен выполнять следующие функции:

– электронный каталог библиографических записей о книгах, журналах, статьях и др., имеющихся у пользователя или заинтересовавших его, т. е. попавших в его поле зрения, в сферу его интересов;

– хранилище цитат из проработанных источников;

– система учета экземпляров книг, журналов и других объектов, хранящихся у пользователя, а также, если это необходимо пользователю, в библиотеках или у других лиц;

– механизм связывания библиографических записей с полнотекстовыми документами в электронной форме (при их наличии), позволяющий организовать быстрый доступ к таким документам.

Использование программного продукта позволит:

– экономить время, традиционно затрачиваемое на поиск библиографической информации и цитат в массиве неупорядоченных файлов и печатных материалов;

– экономить время, требующееся на оформление библиографических списков в рефератах, курсовых работах, научных статьях, диссертациях. Удобнее получать всю информацию библиографического характера из единого центра, только один раз затратив время на корректное оформление каждой записи, вновь вводимой в базу данных электронного архива;

– организовать обмен библиографической информацией с другими студентами, аспирантами, коллегами;

– повысить уровень культуры в работе с библиографической информацией.

Сферы применения программного продукта:

– индивидуальное использование;

– коллективное использование, например, в научном коллективе, занимающемся работами по одной тематике.

**1.3. Ссылки и использованная литература**

**1.3.1. Список документов, упоминаемых в документе-концепции**

1. Архитектура и системные требования (спецификация).

**1.3.2. Список источников, к которым можно обратиться за справками в процессе разработки**

1. Стандарты:

– ГОСТ 7.1–2003 «СИБИД. Библиографическая запись. Библиографическое описание. Общие требования и правила составления»;

– ГОСТ 7.11–2004 «СИБИД. Библиографическая запись. Сокращение слов и словосочетаний на иностранных европейских языках»;

– ГОСТ 7.12–93 «СИБИД. Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила»;

– ГОСТ 7.60–90 «СИБИД. Издания. Основные виды. Термины и определения»;

– ГОСТ 7.80–2000 «СИБИД. Библиографическая запись. Заголовок. Общие требования и правила составления»;

– ГОСТ 7.82–2001 «СИБИД. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления»;

– ОСТ 29.130-97 «Издания. Термины и определения».

2. Система автоматизации библиотек ИРБИС. Общее описание системы [Текст]. – М. : ГПНТБ России, 2002. – 260 с.

**2. Описание пользователей**

**2.1. Виды пользователей и их краткие описания**

**2.1.1. Ученый**

Имеет высокий уровень культуры интеллектуального труда. Читает много разнообразной научной литературы, как периодической, так и непериодической. Пишет научные статьи, доклады для выступления на конференциях и учебно-методические работы.

**2.1.2. Аспирант**

Стремится стать ученым и старается делать то же, что описано в п. 2.1.1.

**2.1.3. Студент**

Объем прорабатываемой научной и учебной литературы – меньше, чем у ученого или аспиранта. Однако объем прочитываемой художественной, научно-популярной и развлекательной литературы может быть большим, чем у ученого, т. к. студент имеет больше свободного времени. Кроме того, студент может иметь в качестве хобби, например, коллекционирование аудиозаписей.

**2.1.4. «Обычный» человек**

Среди людей, не относящихся к категории ученых, также нередко встречаются любители книг, имеющие большие домашние библиотеки. Многие люди накапливают архивы фото-, аудио- и видеоматериалов, которые трудно содержать в порядке без использования компьютерной системы учета.

**2.2. Среда пользователя**

Большинство пользователей используют операционную систему (ОС) Windows, хотя в настоящее время все большую популярность набирает ОС Linux, относящаяся к классу UNIX-подобных ОС.

Зачастую проработанные литературные источники нигде не фиксируются, выписки, сделанные из них, находятся в разрозненных файлах или на бумажных носителях. Для формирования списков использованных источников (библиографических списков), например, при написании реферата или статьи каждый раз используется метод «с нуля». Поэтому пользователям трудно выполнить библиографические описания в соответствии с ГОСТами

**2.3. Основные потребности пользователя**

Общая потребность (возможно, не до конца осознаваемая пользователями в силу разных причин): получить инструмент, позволяющий упорядочить всю библиографическую информацию, которой вынужден оперировать пользователь, и, тем самым, сократить затраты времени на эту деятельность. Также важной потребностью может являться создание электронного каталога домашней библиотеки (включающей не только книги, но также и видео-, фото-, аудиоматериалы).

**2.3.1. Ученый**

Необходима возможность отслеживания выхода новых выпусков периодических изданий и их оперативной проработки, сохранения цитат в базе данных, формирования списков литературы при написании научных статей и учебно-методических работ.

**2.3.2. Аспирант**

То же, что и для ученого.

**2.3.3. Студент**

То же, что и для ученого, с той лишь разницей, что студент не пишет учебно-методических работ, но пишет рефераты, курсовые и дипломные работы.

**2.3.4. «Обычный» человек**

Необходим инструмент, который мог бы помочь в упорядочивании домашней библиотеки, фонотеки, видеотеки.

**3. Состояние рынка и конкурирующие продукты**

**3.1. Характеристика рынка**

На исследуемом рынке представлены два типа программных продуктов:

– «большие» профессиональные автоматизированные библиотечные информационные системы (АБИС), такие, как «ИРБИС», «Академия+» и др., предназначенные для использования в библиотеках;

– «малые» информационные системы, предназначенные для домашнего использования. Последние, как правило, являются программами-каталогизаторами, т. е. позволяют вводить в базу данных описания книг и выполнять выборки из базы данных по различным критериям.

«Большие» АБИС не подходят для персонального использования, т. к. они чрезмерно сложны для пользователя, не имеющего специального библиографического образования. Кроме того, они содержат целый ряд подсистем, предназначенных для выполнения функций, совершенно не нужных в домашних условиях (например, определение книгообеспеченности, списание книг, учет читателей и т. д.). Немаловажным фактором является и тот факт, что подобные системы не являются свободно-распространяемыми – они стоят дорого.

«Малые» же информационные системы, напротив, слишком просты. Они, как правило, хорошо выглядят с технологической стороны: имеют возможности настройки интерфейса пользователя, умеют добывать описания книг на Web-сайтах книжных магазинов (например, Amazon.com), позволяют настроить шрифты, добавить новые таблицы в базу данных и новые поля в таблицы и т. д. Однако с точки зрения соответствия стандартам, принятым в библиотечном деле, эти продукты далеки от совершенства.

Таким образом, необходим программный продукт, условно говоря, среднего класса, который сочетал бы в себе сильные стороны программных продуктов обоих типов, но при этом не был бы слишком сложным в освоении его пользователем, не имеющим специальных знаний.

Потенциальный круг пользователей весьма широк: от ученых до аспирантов и студентов, от работников умственного труда до обычных любителей книги, коллекционирующих также видеофильмы и музыкальные произведения.

Основные технологии, применяемые в отрасли для разработки подобных программных продуктов, следующие: языки программирования – Delphi, C/C++, Visual Basic; СУБД – CDS/ISIS (ЮНЕСКО), Microsoft Access и др. «Малые» информационные системы не работают под управлением операционной системы UNIX (Linux, FreeBSD).

**3.2. Конкурирующие программные продукты**

**3.2.1. Общая характеристика**

К недостаткам существующих «больших» библиотечных информационных систем (при оценке их с позиции индивидуального использования в домашних условиях) можно отнести следующее:

– большое количество функций, не требующихся обычному пользователю;

– сложная система ввода данных, требующая от пользователя специальных знаний в области библиографической деятельности;

– высокая цена программного продукта;

– отсутствие возможности сделать собственные пометки к проработанным источникам (книгам, статьям и т. д.) и сохранить цитаты из этих источников.

Особенностью «больших» библиотечных информационных систем является то, что библиографическое описание в них формируется алгоритмическим способом в соответствии с библиографическими стандартами из элементарных данных, вводимых пользователем.

К недостаткам существующих «малых» информационных систем можно отнести следующее:

– отсутствие возможности формирования библиографических описаний, соответствующих стандартам (ГОСТ 7.1–2003 или аналогичному международному стандарту). Вместо цельного библиографического описания имеется только набор сведений об объектах, внесенных в базу данных: авторы, заглавие, место издания и т. д. В отчетах эти сведения группируются в некое подобие описания, но оно не соответствует никаким стандартам;

– отсутствие развитой возможности конфигурирования программного продукта (с использованием конфигурационного файла или базы данных). Конфигурирование сводится, в основном, к настройке внешнего представления программы и данных, но оно мало помогает в ускорении работы оператора при вводе данных;

– неразвитость системы вспомогательных средств, сокращающих объем работы пользователя при вводе однотипных данных (подобные средства есть во всех Internet-браузерах);

**3.2.2. Система автоматизации библиотек «ИРБИС»**

Разработчик: Государственная публичная научно-техническая библиотека России (ГПНТБ). Сайты: http://www.gpntb.ru, http://www.elnit.org.

«Большая» библиотечная информационная система. Очень мощная, сложная, дорогая.

Фрагмент описания программного продукта, взятый с сайта разработчика (стиль и орфография сохранены)

Основные характеристики:

– поддержка произвольного количества баз данных, составляющих Электронный каталог или представляющих собой проблемно-ориентированные библиографические базы данных;

– технология автоматического формирования словарей, на основе которых реализуется быстрый поиск по любым элементам описания и их сочетаниям;

– средства для ведения и использования Авторитетных файлов, баз данных УДК, ББК, ГРНТИ и Тезауруса;

– поддержка традиционных «бумажных» технологий: от печати форм заказа/подписки и листов книги суммарного учета до печати всех видов каталожных карточек;

– технологии, ориентированные на использование штрих-кодов и радиометок на экземплярах изданий и читательских билетах;

– поддержка многоязычия на основе UNICODE, т. е. возможность ввода на любых языках мира;

– поддержка ссылок от библиографических описаний на полные тексты, графические данные и другие внешние объекты (включая ресурсы Интернет);

– средства для создания и ведения полнотекстовых баз данных (электронной библиотеки);

– специальные средства для создания имидж-каталогов по ретрофонду библиотеки на основе графических образов каталожных карточек и автоматического распознавания их текстов;

– средства для перевода пользовательских интерфейсов на другие языки;

– широкий набор сервисных средств, обеспечивающих удобство и наглядность пользовательских интерфейсов, упрощающих процесс ввода, исключающих ошибки и дублирование информации;

– широкие возможности для адаптации к условиям работы конкретной библиотеки, включая средства создания уникальных рабочих профилей для всех категорий пользователей;

– открытость, позволяющая пользователю самостоятельно вносить изменения в широких пределах: от изменения входных и выходных форм до разработки оригинальных приложений.

«Оригинальное программное обеспечение системы написано на Delphi с использованием библиотеки ISIS32.DLL (Bireme, Бразилия). Физическая структура БД соответствует СУБД CDS/ISIS (ЮНЕСКО)» (цитата из описания программного продукта).

**3.2.3. Автоматизированная библиотечная информационная система «Академия+»**

Разработчик: Центр автоматизированных технологий «Ростехноком». Сайты: http://www.rostechnocom.ru; http://www.academy-plus.ru.

«Большая» библиотечная информационная система. Очень мощная, сложная, дорогая.

Фрагмент описания программного продукта, взятый с сайта разработчика (стиль и орфография сохранены)

Основные характеристики:

– масштабируемая трехуровневая клиент-серверная архитектура;

– независимость от программно-аппаратной платформы (любая аппаратная платформа: IBM PC, SUN и др.; любая операционная система: Windows, UNIX, LINUX);

– независимость от СУБД (любая реляционная СУБД: ORACLE, MS SQL Server, My SQL, PostgresSQL и др.);

– работа системы в Internet и Intranet без ограничения количества пользователей;

– поддержка UNICODE и штрих-кодирования на программном уровне.

**3.2.4. BookCAT**

Разработчик: FNProgramvare (Норвегия). Сайт: http://www.fnprg.com.

«Малая» информационная система. Самая развитая из систем подобного класса.

**3.2.5. Учет книг**

Разработчик: «Простой Софт». Сайт: http://www.simple-soft.ru, http://www.prostoysoft.ru.

Фрагмент описания программного продукта, взятый с сайта разработчика (стиль и орфография сохранены)

**Основные функции программы**

**Ведение базы книг, журналов. Каталогизация**

В базе данных содержится информация о книгах, журналах. Предусмотрены такие поля как – название, авторы, категория, тип, издательство, серия, формат, год издания, количество страниц, тираж, обложка, ISBN, УДК, № шкафа, № полки, блок, подблок, время добавления и т. д. Для каждой книги показываются все ее читатели (которые читали эту книгу ранее и читают сейчас).

Предусмотрены удобные способы сортировки и фильтрации данных, что позволяет быстро найти нужные книги. Любую таблицу базы можно распечатать, экспортировать в MS Word, MS Excel или текстовый формат CSV. Имеется импорт из других источников данных в формате CSV.

**Учет должников по возврату книг, журналов**

Система фиксирует информацию о читателях – ФИО, контактная информация, выданные книги, даты выдачи и возврата книг. Контролируя значение поля с датой возврата книги можно легко вести учет должников. Таблица «На руках» показывает список всех выданных на руки книг и журналов.

**Функциональные возможности программы**

С помощью программы вы сможете делать следующее:

Создавать, изменять, удалять записи, поля, таблицы.

Импортировать данные в любую таблицу базы данных из текстовых файлов.

Удалять дублированные записи с одинаковым названием и автором. Можно настроить по-другому.

Сортировать таблицы по любому полю, включая сортировку по нескольким полям (до 3-х) удерживая клавишу Shift.

Фильтровать таблицу по любому полю, используя следующие операторы: =, >, >=, <, <=, <>, «Содержит», «Не содержит», «Начинается с», «Не начинается с», «Кончается на», «Не кончается на», LIKE, NOT LIKE.

Группировать одинаковые данные в любом поле, когда таблица отсортирована по этому полю (для отмеченных полей в свойствах таблицы).

Помечать записи как «Избранное», тогда они будут отображаться оранжевым цветом. Цвет задается в свойствах таблицы.

Помечать записи как «Мертвое» («Неинтересное»), тогда они будут отображаться серым (или другим) цветом.

Настраивать правила цветовыделения. Вы сами определяете, какие строки, каким цветом и при каких условиях выделять.

Строить дерево по любым полям с произвольным количеством уровней для иерархического отображения данных любой таблицы.

Изменять данные в любом поле (кроме ID и вычисляемых полей) прямо в таблице или в отдельной форме (выбирается в настройках), отмечать несколько записей, удалять, печатать, экспортировать отмеченные.

Изменять или удалять сразу несколько записей в любой таблице базы данных с помощью формы «Групповое обновление».

Создавать новые хранимые поля для таблиц следующих типов: текстовое, числовое, Да/Нет, Дата и время.

Создавать вычисляемые поля для таблиц, например можно создать поле с формулой «[Поле 1] / [Поле 2]».

Создавать вычисляемые поля, значения которых будут браться из других таблиц. Например, можно вывести имя должника из таблицы «Должники».

Создавать новые таблицы с абсолютно такими же возможностями по действиям с ними, как и у любой другой таблицы

Привязывать ниспадающие списки полей к другим таблицам для легкого выбора значений из них при редактировании в таблице или для выбора из других форм при редактировании в форме.

Задавать произвольное количество подчиненных таблиц для любой таблицы, для чего необходимо задать привязку по полям в свойствах таблицы.

Менять порядок следования полей в любой таблице, используя перетаскивание или с помощью формы «Настройки».

Переименовывать поля таблиц и названия самих таблиц в соответствии со спецификой вашего бизнеса. (Точнее говоря, их лейблы.)

Печатать текущее представление любой таблицы с учетом видимости полей, их ширины и порядка.

Экспортировать данные любой таблицы в MS Excel или текстовый CSV-файл с учетом текущего представления таблицы.

Экспортировать текущую запись в MS Word на основе файла-шаблона с закладками, соответствующими названиям полей.

Работать с несколькими файлами баз данных, создавать новые базы данных, разумеется, можно также открывать их с помощью MS Access.

**3.2.6. eLibPro**

Разработчик: Songs Technologies. Сайт: http://songstech.com.

«Малая» информационная система. Использует формат данных mdb (Microsoft Access).

**3.2.7. Librarian Pro**

Разработчик: Koingo Software. Сайт: http://www.koingosw.com.

«Малая» информационная система.

**3.2.8. HomiStorage**

Разработчик: Hominoid Software. Сайт: http://www.hominoid.nm.ru.

«Малая» информационная система.

**3.2.9. All My Books**

Разработчик: Bolide Software. Сайт: http://www.bolidesoft.com.

«Малая» информационная система.

**3.3. Определение позиции продукта на рынке**

Для ученых, аспирантов, студентов и всех других пользователей,

которые хотят иметь возможность упорядочить свою коллекцию книг, журналов, видео- и аудиоматериалов, рукописей и электронных документов.

«Электронный архив» является автоматизированной информационной системой для управления домашней библиотекой.

Наш продукт позволит всегда иметь под рукой качественную библиографическую информацию.

В отличие от конкурирующих «малых» информационных систем, таких, как BookCAT, «Учет книг» и др.,

наш продукт разработан с учетом требований стандартов на библиографическое описание, использует «большую» СУБД, имеет развитые средства помощи пользователю для ускорения ввода данных.

**4. Атрибуты функций**

**4.1. Приоритет**

*Критический*. Основные функции. Если их не удастся реализовать, сис­тема не будет удовлетворять потребности заказчика. В версии должны быть реализованы все критические функции, в противном случае график являет­ся нереальным.

*Важный*. Функции, важные для успешной и эффективной работы системы в большинстве приложений. Данные функциональные возможности нельзя легко обеспечить иным способом. Если важные функции не войдут в реали­зацию, это может повлиять на удовлетворение пользователя или заказчика результатом работы или даже на доходы от продаж, но выпуск версии не должен задерживаться из-за нехватки некой важной функции.

*Полезный*. Функции, которые нужны в менее распространенных прило­жениях, будут использоваться не так часто или их можно достаточно эф­фективно Заменить другими действиями. Если они не войдут в реализацию,это не окажет заметного воздействия на отношение заказчика или доходы.

**4.2. Уровень трудозатрат**

*Низкий*. Требуется не более 1 человеко-недели на разработку.

*Средний*. Требуется от 1 до 4 человеко-недель на разработку.

*Высокий*. Требуется более 4 человеко-недель на разработку.

**4.3. Риск**

Указывается симптомы, по которым функция может быть отнесена к соответствующему уровню риска и возможные последствия при реализации риска.

*Низкий.* Назначение хорошо понято. Ресурсы требуемой квалификации доступны в полном объеме. Разрабатываемые подсистемы не потребуют новой технологической платформы. Превышение сроков и бюджета при реализации функции маловероятно.

*Средний*. Назначение и рамки определены более-менее четко. Ресурсы требуемой квалификации доступны в основном. Подсистемы создаются на новой, но стабильной технологической платформе. Последствия — функция может быть реализована с превыщением бюджета или сроков не более 1,5-2 раз.

*Выше среднего*. Назначение и рамки определены недостаточно четко. Ресурсы требуемой квалификации сильно ограничены. Подсистемы создаются на новой технологической

платформе, сомнения в рыночной стабильности платформы. Последствия — функция может быть реализована с превыщением бюджета или сроков в 2 или более раз.

*Высокий*. Назначение и рамки определены нечетко. Ресурсы требуемой квалификации практически отсутствуют. Подсистемы создаются на новой технологической платформе, в отношении которой крайне мало ясности. Последствия — функция может быть не реализована.

**4.4. Целевая версия**

Записывается, в какой версии продукта предполагается впервые реализовать данную функцию.

**5. Функциональные требования (функции продукта)**

**5.1. Обязательные функции для первой версии**

Название «первая» является условным. Номер версии должен быть определен на основе фактического уровня готовности программного продукта. Предположительно это будет версия 0.5.

**ПФТ1. Соответствие требованиям ГОСТ 7.1–2003**

1. Архитектура программного продукта и структура базы данных должны быть разработаны с учетом требований ГОСТ 7.1–2003, а именно: с учетом наличия определенных областей библиографического описания и элементов этих областей. Причем, необходимо учесть, что ряд элементов в библиографическом описании могут повторяться (например, сведения об ответственности, место издания, примечание и др.).

2. Должна быть обеспечена возможность хранения в базе данных библиографических записей о книгах, статьях, электронных ресурсах, научных конференциях и т. д.

*Обоснование.* Указанный ГОСТ является основным стандартом, регламентирующим создание библиографических описаний любых объектов: книг, журналов, диссертаций, материалов конференций, статей, нот, карт, рукописей, патентных документов, электронных ресурсов и т. д.

*Ссылки на спецификацию.* Пп. 3.1, 3.9.

**ПФТ2. Ввод библиографических описаний в базу данных в готовом виде**

1. Библиографические описания должны вводиться в базу данных сразу в готовом виде, а не формироваться из элементарных фрагментов библиографического описания, как это делается в «больших» АБИС (например, «ИРБИС»). Таким образом, библиографическое описание формируется пользователем, а не программным продуктом.

2. Должна быть предусмотрена возможность ввода в базу данных детальных сведений, соответствующих областям и элементам областей библиографического описания (заглавие, сведения об ответственности и т. д.). Детальные сведения могут быть введены позднее, если пользователь нуждается в этом, но могут и не вводиться совсем.

*Обоснование.* Предлагаемая схема действий является инверсией по отношению к традиционной схеме. Такая схема позволит реализовать различные способы использования программного продукта. Простейший способ предусматривает только ввод библиографических описаний в целом, без ввода детальных данных по каждой области библиографического описания. Такой подход упрощает и ускоряет работу на начальном этапе, но в этом случае пользователь не сможет выполнять целый ряд поисковых операций в базе данных.

*Ссылки на спецификацию.* П. 3.9.

**ПФТ3. Обработка наличия различных вариантов имен собственных**

1. База данных должна быть спроектирована таким образом, чтобы наличие различных вариантов написания фамилий персоналий и наименований организаций, переименований городов и т. п. не препятствовало однозначной идентификации таких объектов пользователем.

2. Пользователь должен иметь возможность при выборе любого из вариантов, представленных в базе данных, получить выборку, включающую и все другие варианты.

*Обоснование.* Подобные разночтения фамилий персоналий могут появиться, например, при переводе иностранных книг на русский язык (*Вильямс* и *Уильямс*). Поэтому если пользователь производит поиск книг автора по фамилии *Вильямс*, то в выборку должны быть включены и книги этого же автора, но в переводе представленного как *Уильямс*, а в оригинале – как *Williams*. При этом в выборку не должны попасть книги однофамильцев, поскольку это уже различные персоналии.

*Ссылки на спецификацию.* Пп. 3.7, 3.9.

**ПФТ4. Фиксирование результатов проработки источников**

1. Должна быть предусмотрена возможность фиксирования даты и результатов проработки источников (книг, журналов, статей и т. д.).

2. Должна быть предусмотрена возможность выставления оценок проработанным источникам. Пользователь должен иметь возможность сам назначить показатели, по которым выставляются эти оценки.

3. Пользователь должен иметь возможность ввести в базу данных содержание (оглавление) проработанного источника.

*Обоснование.*

*Ссылки на спецификацию.* П. 3.9.

**ПФТ5. Хранение выписок из проработанных источников**

1. Должна быть предусмотрена возможность хранения выписок в базе данных.

2. Должна быть предусмотрена возможность связывания таких выписок, которые могут храниться также и в файлах, имеющих различные форматы (Microsoft Word, HTML, ASCII), с соответствующими библиографическими описаниями.

*Обоснование.* Необходимо избавить пользователя от утомительного поиска нужного файла методом визуального просмотра каталогов операционной системы (как зачастую бывает в действительности).

*Ссылки на спецификацию.* П. 3.9.

**ПФТ6. Система ключевых слов**

1. Должна быть предусмотрена система ключевых слов.

2. Должна быть предусмотрена возможность проведения выборки библиографических описаний для каждого ключевого слова.

*Обоснование.*

*Ссылки на спецификацию.* П. 3.9.

**ПФТ7. Простые средства поиска информации**

1. Должны быть предусмотрены простые средства поиска в базе данных и выборки информации из нее. На последующих этапах разработки эти средства должны быть значительно усилены.

*Обоснование.*

*Ссылки на спецификацию.* Пп. 3.2, 3.7.

**ПФТ8. Учет экземпляров**

1. Должна быть предусмотрена возможность описания экземпляров книг, журналов, дисков, рукописей и т. д., находящихся как в собственности пользователя, так и в библиотеках или у других лиц.

*Обоснование.*

*Ссылки на спецификацию.* П. 3.9.

**5.2. Дополнительные функции для первой версии**

**ПФТ9. Формирование тематических библиографических списков**

1. Должна быть предусмотрена возможность формирования библиографических списков, включающих описания группы объектов (книг, статей и т. д.), содержащихся в базе данных.

2. Списки должны формироваться без повторного ввода библиографических описаний в базу данных (без дублирования данных в ней).

*Обоснование.* Такие списки могут включать, например, литературу, предназначенную для выполнения одной задачи или характеризующую одну тему, или относящуюся к одному периоду времени (статьи за конкретный год по конкретной тематике) и т. п. При подготовке научной статьи или реферата ученый или студент сможет включить в такой список соответствующие источники и затем перенести сформированный список в статью или реферат в качестве перечня цитируемых источников.

*Ссылки на спецификацию.* П. 3.9.

**ПФТ10. Развитые средства помощи пользователю для ускорения ввода данных**

1. Должен быть создан механизм для ускоренного ввода типовых фрагментов текста в поля библиографического описания.

*Обоснование.* Такими типовыми фрагментами могут быть: фамилии авторов, сведения, относящиеся к заглавию, названия издательств, тексты примечаний и т. д.

*Ссылки на спецификацию.* П. 3.2, 3.7.

**5.3. Будущие функции**

**ПФТ11. Развитые средства поиска информации**

1. Должны быть предусмотрены различные средства поиска в базе данных и выборки информации из нее по различным критериям.

*Обоснование.*

*Ссылки на спецификацию.* П. 3.2, 3.7.

**ПФТ12. Иерархическая система ключевых слов**

1. Система ключевых слов должна быть организована по иерархическому принципу.

2. Должна быть предусмотрена возможность выполнения поиска информации в соответствии с требуемым уровнем иерархии ключевых слов.

*Обоснование.* Реализация этого требования позволит пользователю регулировать объем выбираемых библиографических описаний и точность соответствия запросу. Например, для ключевого слова «Linux» словом-родителем будет ключевое слово «операционная система», а для него, в свою очередь, таким словом-родителем будет «система». Отметим, что в известных нам программных продуктах, используемых в библиотеках, иерархический подход к организации системы ключевых слов не реализован.

*Ссылки на спецификацию.* П. 3.7, 3.9.

**ПФТ13. Формирование библиографического описания из элементарных данных**

1. Должен быть реализован и традиционный способ алгоритмического формирования библиографического описания из элементарных данных, предварительно введенных пользователем в области и элементы библиографического описания (заглавие, сведения об ответственности, издатель, место издания, год издания и т. д.).

*Обоснование.* Реализация этого требования позволит опытному пользователю ускорить процесс ввода сведений в базу данных при полномасштабном использовании программного продукта.

*Ссылки на спецификацию.* П. 3.7.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Функция | Приоритет | Уровень трудозатрат | Риск | Целевая версия |
| ПФТ1 | критический | средний | низний | 1.0 |
| ПФТ10 | важный | высокий | средний | 1.0 |
| ... | ... |  |  |  |
| ПФТ13 | полезный | средний | низкий | 3.0 |

**6. Основные способы использования программного продукта и сценарии работы с ним**

**6.1. Простой способ использования программного продукта**

Пользователь вводит в базу данных только библиографические описания (полные и краткие либо только краткие). Он не использует систему справочников. Он не вводит в базу данных детальные сведения, содержащиеся в областях библиографического описания (заглавие, сведения об ответственности, издатель, место издания, год издания и т. д.). В таком случае пользователь лишен возможности выполнять б*о*льшую часть процедур поиска сведений в базе данных.

**6.2. Полномасштабный способ использования программного продукта**

Пользователь использует систему справочников, дополняет их новыми данными по мере необходимости. Он вводит в базу данных полные и краткие библиографические описания (либо только краткие, если описание выполняется не на основе оригинального объекта, а на основе вторичной информации, например, библиографического списка в книге, диссертации, статье). Он вводит в базу данных детальные сведения, содержащиеся в областях библиографического описания (заглавие, сведения об ответственности, издатель, место издания, год издания и т. д.). Также пользователь ведет учет экземпляров (книг, журналов, дисков, рукописей и т. д.), как находящихся в собственности пользователя, так и находящихся в библиотеках или у других лиц.

**7. Нефункциональные требования**

**7.1. Требования к программному продукту**

**7.1.1. Требования к инсталляции**

Необходимо разработать процедуру инсталляции программного продукта.

**7.1.2. Требования к эксплуатации**

7.1.2.1. Требования к удобству эксплуатации (практичность)

*7.1.2.1.1. Необходимое время подготовки пользователя для достижения минимальной производительности*

Приблизительно 1–2 дня. Пользователю необходимо ознакомиться с ГОСТ 7.1–2003. Он может ограничиться только просмотром примеров библиографических описаний, приведенных в Приложении к этому ГОСТу.

*7.1.2.1.2. Время выполнения типичных задач или транзакций, осуществляемых пользователем*

Ввод полного и краткого библиографических описаний для одной книги (статьи, диска и т. д.) должен занимать не более 10 минут. Ввод дополнительных сведений, в соответствии с областями библиографического описания, должен занимать не более 1 часа. По мере повышения квалификации пользователя эти временн*ы*е интервалы должны уменьшиться до 5 минут и 20 минут соответственно. Это должно быть достигнуто за счет того, что по мере приобретения навыков работы с программным продуктом пользователь должен постепенно заполнить различные справочники, содержащиеся в базе данных, а также настроить конфигурационный файл, содержащий типовые фрагменты библиографических описаний (имена и отчества авторов, наименования часто встречающихся издательств и т. п.).

*7.1.2.1.3. Сравнение практичности новой системы с уже существующими современными системами*

Практичность должна быть не хуже, чем у всех систем, перечисленных в документе-концепции.

*7.1.2.1.4. Следование соглашениям и стандартам, разработанным для человеко-машинного интерфейса*

Пользовательский интерфейс программного продукта должен отвечать, как минимум, требованиям здравого смысла. Хотя это требование неконкретное, но, поскольку автором документа-концепции и разработчиком будет одно и то же лицо, а также в силу ограниченности времени на разработку, особых требований относительно соответствия стандартам, разработанным для человеко-машинного интерфейса, на данном этапе не предъявляется. В дальнейшем возможны изменения технологии создания пользовательского интерфейса, например, использование каскадных таблиц стилей (CSS).

*7.1.2.1.5. Прочие требования к удобству эксплуатации*

**ПНТ1. Экспорт и импорт данных**

1. Должна быть реализована возможность экспорта данных из системы в виде команд языка SQL и импорта данных, представленных в виде таких команд.

*Обоснование.* Это позволит организовать обмен данными между пользователями предлагаемой системы. Поскольку оформление библиографических записей представляет собой трудную задачу для большинства студентов и аспирантов, то в процессе заполнения индивидуальной базы данных каждого пользователя важную роль может сыграть взаимный обмен данными. В частности, преподаватель, рекомендуя студентам перечень источников для изучения материала конкретной учебной дисциплины, мог бы выдавать им этот перечень сразу в той форме, которая позволяет выполнить его импорт в базу данных каждого студента.

*Ссылки на спецификацию.* П. 3.10.

**ПНТ2. Конфигурирование программного продукта**

1. Должна быть предусмотрена возможность конфигурирования программного продукта.

2. Конфигурационные параметры должны описывать, в основном, значения полей документов, принимаемые по умолчанию и позволяющие пользователю ускорить процесс ввода данных.

3. В первой версии программного продукта реализация конфигурирования может быть выполнена на основе использования текстового конфигурационного файла, изменения в который должны вноситься пользователем в текстовом редакторе.

4. В последующих версиях должна быть разработана интерактивная система конфигурирования программного продукта.

*Обоснование.* Реализация этого требования позволит пользователю ускорить процесс ввода данных, сделать его более удобным и уменьшить число ошибок ввода.

*Ссылки на спецификацию.* П. 3.8.

7.1.2.2. Требования к режиму эксплуатации

*7.1.2.2.1. Доступность (сколько дней в неделю и часов в сутки должна быть доступна система)*

Эпизодически, по мере возникновения потребности.

*7.1.2.2.1. Прочие требования к режиму эксплуатации*

Нет требований.

7.1.2.3. Прочие требования к эксплуатации

Нет требований.

**7.1.3. Требования к эффективности**

7.1.3.1. Требования к производительности

*7.1.3.1.1. Время ответа для транзакции (среднее, максимальное)*

Время получения результатов при выполнении выборки библиографических описаний не должно превышать 2–3 секунды.

*7.1.3.1.2. Пропускная способность (число транзакций в секунду)*

Нет требований.

*7.1.3.1.3. Емкость (число пользователей или транзакций, которые может обслужить система)*

Нет требований.

*7.1.3.1.4. Режимы снижения производительности (допустимые режимы работы при ухудшении параметров системы)*

Нет требований.

*7.1.3.1.5. Время обновления экрана*

Нет требований.

*7.1.3.1.6. Время реакции на действия пользователя*

Нет требований, т. к. все операции по вводу и корректировке данных не связаны с обработкой больших объемов данных.

7.1.3.2. Требования к ресурсам

*7.1.3.2.1. Клиентский компьютер*

Оперативная память: объем, достаточный для нормальной работы Web-браузера. Объем дискового пространства: нет требований. Процессор: любой современный процессор.

*6.1.3.2.2. Сервер*

Оперативная память: объем, достаточный для нормальной работы Web-сервера и сервера баз данных. Объем дискового пространства: приблизительно 100–150 Мб с учетом СУБД PostgreSQL (дополнительный требуемый объем зависит от объема электронных документов). Процессор: определяется требованиями, которые предъявляет СУБД PostgreSQL (см. документацию на текущую версию этой СУБД).

7.1.3.3. Прочие требования к эффективности

Нет требований.

**7.1.4. Требования к надежности**

7.1.4.1. Вероятность отказа

Нет требований, т. к. от программного продукта не требуется непрерывное функционирование.

7.1.4.2. Частота отказов

Нет требований, т. к. от программного продукта не требуется непрерывное функционирование.

7.1.4.3. Среднее время безотказной работы (среднее время между двумя последовательными сбоями)

Не менее отрезка времени, требующегося для ввода данных об одном объекте библиографического описания (книге, статье и т. д.).

7.1.4.4. Вероятность готовности системы к использованию

Нет требований, т. к. от программного продукта не требуется непрерывное функционирование.

7.1.4.5. Среднее время восстановления после сбоя (отказа)

Нет требований, т. к. программный продукт не является критическим по данному показателю.

7.1.4.6. Вероятность порчи данных при сбое (отказе)

Нет требований, т. к. предполагается, что СУБД обеспечивает откат незавершенных транзакций, гарантируя тем самым целостность базы данных.

7.1.4.7. Прочие требования к надежности

Нет требований.

**7.1.5. Требования к переносимости**

7.1.5.1. Поддерживаемые операционные системы

*7.1.5.1.1. Клиентский компьютер*

Операционные системы UNIX (FreeBSD, Linux) и Windows (XP/2003/Vista/7). На клиентском компьютере работает Web-браузер.

*7.1.5.1.2. Сервер*

Операционная система UNIX (FreeBSD, Linux). В будущем предполагается перенос продукта в среду операционной системы Windows (XP/2003/Vista/7).

7.1.5.2. Объем машинно-зависимых операторов и подсистем

Таких операторов и подсистем нет.

7.1.5.3. Прочие требования к переносимости

Нет требований.

**7.1.6. Требования к защищенности**

**ПНТ3. Аутентификация пользователей**

1. При входе пользователя в систему должна выполняться аутентификация.

*Обоснование.* Это необходимо для организации разграничения полномочий доступа к базе данных.

*Ссылки на спецификацию.* Пп. 3.1, 3.9.

**ПНТ4. Интерактивная система управления разграничением полномочий доступа к базе данных**

1. Должна быть разработана интерактивная система для управления разграничением полномочий доступа различных пользователей к базе данных.

*Обоснование.*

*Ссылки на спецификацию.* Пп. 3.5, 3.9.

**ПНТ5. Ведение журнала операций с базой данных**

1. Должна быть предусмотрена возможность ведения журнала всех операций с базой данных, изменяющих ее состояние: ввод записей, их обновление и удаление.

2. Должна быть предусмотрена возможность просмотра истории всех операций с базой данных с применением специального пользовательского интерфейса.

*Обоснование.* В случае ошибочного изменения или удаления данных пользователь будет иметь возможность «реанимировать» их.

*Ссылки на спецификацию.* П. 3.9.

**7.1.7. Требования к безопасности**

При инсталляции программного продукта необходимо сохранять резервные копии всех конфигурационных файлов, принадлежащих другим программным продуктам, в случае внесения изменений в эти файлы.

**7.1.8. Требования к удобству сопровождения**

Должны быть подробные комментарии в исходных текстах программ.

**7.1.9. Требования к точности числовых вычислений**

Нет требований.

**7.1.10. Прочие требования к программному продукту**

**ПНТ6. Организация коллективной работы**

1. Должна быть предусмотрена возможность коллективного использования программного продукта через локальную сеть или Internet, а также на одном компьютере.

2. Должен быть организован раздельный доступ к данным, принадлежащим различным пользователям.

*Обоснование.*

*Ссылки на спецификацию.* Пп. 3.1, 3.9.

**7.2. Организационные требования**

**7.2.1. Выходные требования**

7.2.1.1. Сроки разработки и изготовления

Первый работающий вариант программного продукта должен быть готов к 15 сентября 2006 г.

7.2.1.2. Сопутствующая документация

Нет сопутствующей документации.

7.2.1.3. Прочие выходные требования

Нет требований.

**7.2.2. Требования к реализации**

7.2.2.1. Модель организации разработки

Модель пошаговой разработки.

7.2.2.2. Методы проектирования и документирования разработки

Нет требований.

7.2.2.3. Языки программирования и инструментальные средства

1. Языки – Perl (версия 5.8.x) и JavaScript; СУБД – PostgreSQL (версия 8.x); Web-сервер – Apache (версия 2.2.x). Инструментальные средства: отладчик языка Perl; отладчик языка JavaScript, встроенный в Web-браузер. Библиотеки: интерфейс к СУБД PostgreSQL – pgperl (до версии 7.3 входил в состав дистрибутива СУБД PostgreSQL).

2. Рассмотреть также возможность использования системы управления версиями программного продукта (Subversion, Git и др.).

3. Рассмотреть возможность использования общего интерфейса DBI/DBD к СУБД PostgreSQL.

7.2.2.4. Локализация и интернационализация

Необходимо реализовать поддержку локализации (l10n) и интернационализации (i18n) на более поздних этапах разработки. Поддерживаемые языки: русский и английский.

**7.2.3. Требования к используемым стандартам**

1. Использовать стандарты кодирования и оформления исходных текстов программ, аналогичные стандартам проекта GNU.

2. Комментарии в исходных текстах программ должны быть написаны на русском языке.

**7.2.4. Требования к лицензированию**

Программный продукт должен распространяться по лицензии, аналогичной лицензии GNU GPL.

**7.2.5. Требования к дистрибуции (распространению) и вопросы цены**

1. Распространение продукта будет производиться, в основном, через Web-сайт http://www.morgunov.org.

2. Программный продукт будет распространяться бесплатно в исходных текстах.

**7.2.6. Вопросы авторских прав**

Авторские права принадлежат Моргунову Е. П.

**7.3. Внешние требования**

**7.3.1. Требования к взаимодействию с другими системами**

1. Необходимо организовать взаимодействие с СУБД PostgreSQL, которая будет использоваться для хранения данных, и с Web-сервером Apache.

2. Взаимодействие с другими *прикладными* программными продуктами не предполагается.

**7.3.2. Юридические требования**

Для разработки программного продукта должны использоваться только инструменты с лицензией, разрешающей их бесплатное использование.

**7.3.3. Этические требования**

Нет требований.

**7.4. Прочие требования**

Нет требований.

**8. Требования к документации**

**8.1. Руководство пользователя**

1. Должно содержать:

– основы методики составления библиографического описания (на основе ГОСТ 7.1–2003);

– примеры библиографических описаний;

– описание процедур работы с продуктом;

– глоссарий;

– индекс для поиска по тексту.

2. Форма руководства – электронная, формат – pdf, html. Должна быть предусмотрена возможность вывода руководства на печать (полностью или отдельных разделов).

**8.2. Руководство программиста**

Должно содержать следующие сведения (но не обязательно ограничиваться ими):

– описание порядка установки программного продукта и его первичной настройки;

– описание модулей программы и их взаимосвязей;

– описание типовой структуры модуля и порядка его функционирования.

**8.3. Интерактивная подсказка**

1. Должна быть контекстно-зависимой и иметь развитую систему ссылок и функцию поиска по тексту.

2. Должна быть в состоянии предложить пользователю подсказки по заполнению полей экранных форм.

**8.4. Руководства по инсталляции и конфигурированию и файл ReadMe**

Должны быть представлены в электронной форме (формат файла – ASCII). Файл ReadMe должен содержать перечень изменений, проведенных в новой версии, а также список замеченных ошибок.

**8.5. Маркировка и упаковка**

В процессе инсталляции необходимо выводить информацию о версии программного продукта и о его авторе. При распространении продукта на CD-дисках предусмотреть наличие маркировки с информацией о наименовании и версии продукта, а также сведений об авторе.

**9. Глоссарий и список сокращений**

**9.1. Глоссарий**

**9.2. Список сокращений**

**АБИС** – автоматизированная библиотечная информационная система

**ББК** – библиотечно-библиографическая классификация

**БД** – база данных

**ГПНТБ** – Государственная публичная научно-техническая библиотека России

**ГРНТИ** – Государственный рубрикатор научно-технической информации

**ОС** – операционная система

**ПНТ** – пользовательское нефункциональное требование

**ПФТ** – пользовательское функциональное требование

**СИБИД** – Система стандартов по информации, библиотечному и издательскому делу

**СУБД** – система управления базами данных

**УДК** – универсальнаядесятичная классификация (система классификации информации, широко используется во всем мире для систематизации произведений науки, литературы и искусства, периодической печати, различных видов документов и организации картотек)

**ЮНЕСКО** (англ. UNESCO – United Nations Educational, Scientific and Cultural Organization) – Организация Объединённых Наций по вопросам образования, науки и культуры

**CSS** – Cascading Style Sheets (каскадные таблицы стилей)

**CSV** – Comma separated values (текстовый формат, предназначенный для представления табличных данных)

**GNU** – рекурсивный акроним от англ. GNU’s Not UNIX (проект, основанный Ричардом Столлманом – http://www.gnu.org)

**GNU GPL** – GNU General Public License (лицензия на свободное программное обеспечение, созданная в рамках проекта GNU)

**I18n** – internationalization (интернационализация)

**ISBN** – International Standard Book Number (международный стандартный номер книги – уникальный номер книжного издания, необходимый для распространения книги в торговых сетях и автоматизации работы с изданием)

**L10n** – localization (локализация)