Аннотация

Настоящий документ является пояснительной запиской курсового проекта на тему «Разработка программы для учёта приёма таблеток». Пояснительная записка включает в себя введение, составленные в соответствии с требованиями ГОСТ ЕСПД программные документы «Описание программы» и «Руководство оператора», заключение, список литературы и приложения. В приложениях помещены программный код, схемы по ГОСТ ЕСПД, иллюстрации процесса разработки и функционирования программного продукта.

Содержание

|  |  |
| --- | --- |
| Введение |  |
| 2 Специальная (практическая, опытно-экспериментальная) часть |  |
| 2.1 Анализ задачи |  |
| 2.2 Описание программы |  |
| 2.3 Руководство оператора |  |
| Заключение |  |
| Список используемой литературы |  |
| Приложение А. Информационная модель предметной области |  |
| Приложение Б. Текст программы |  |
| Приложение В. Иллюстрации процесса разработки программы |  |
| Приложение Г. Схемы |  |
| Приложение Д. Иллюстрации запуска и работы программы |  |

Введение

Важным направлением современного программирования является разработка настольного программного обеспечения. Такие программы позволяют пользователям решать задачи, связанные с обработкой данных в определенных областях деятельности. Например, для обработки фотографий пользователь может воспользоваться графическими редакторами, а для прослушивания музыки или просмотра видео – мультимедийные проигрыватели.

Задачей данного курсового проекта является создание приложения для удобного учёта приёма таблеток. Пользователь может добавлять, удалять, редактировать список лекарств, отмечать количество принятых таблеток, отмечать необходимое количество таблеток в день, указывать даты начала и окончания приёма таблеток. Это значит, что пользователь сможет удобно вести учёт приёма лекарств. Также приложение будет многопользовательским – это значит, что им смогут пользоваться несколько людей, у каждого из которых будет свой список лекарств. Для этого необходимо будет реализовать авторизацию пользователей в приложении. Таким образом, тема дипломного проекта актуальна.

2 Специальная (практическая, опытно-экспериментальная) часть

2.1 Анализ задачи

Задачей данного курсового проекта является создание приложения для учёта приёма таблеток.

Так как программа является многопользовательской, то при запуске приложения появляется окно с авторизацией в том случае, если пользователь уже есть в базе данных и возможность регистрации, если пользователь только начинает использовать программу.

После авторизации или регистрации пользователь попадает на главное меню, где его приветствует электронный ассистент. В главном меню пользователь может выбрать одно из следующих действий:

* просмотр, добавление, редактирование, удаление лекарства из списка;
* просмотр информации о дате приёма и количестве принятых таблеток;
* добавление даты и количества принятых таблеток;
* покупка лекарств;
* поиск и добавление таблеток в базу;
* рекомендация лекарств по категориям.

Разработку программы начнём с построения информационной модели предметной области. В данном случае предметной областью является реальный мир. Чтобы построить информационную модель, необходимо решить, какие его объекты и какие свойства этих объектов нас будут интересовать.

Три основных объекта предметной области – лекарства, пользователь и список лекарств для каждого отдельного пользователя. Также можно выделить такие дополнительные объекты, как: категория лекарства (например, лекарство от головной боли), список приёма лекарств для каждого отдельного пользователя.

Таким образом, получаем следующий список объектов:

* пользователи;
* лекарства;
* список лекарств;
* категория лекарства;
* список приёма лекарств.

Теперь следует определить свойства этих объектов.

Для объекта пользователь нас интересуют следующие свойства:

* код пользователя;
* имя;
* пол;
* логин;
* пароль.

Объект лекарство характеризуется следующими атрибутами:

* код лекарства;
* название;
* категория.

Связь многие-ко-многим между пользователями и лекарствами реализует вспомогательная сущность список лекарств со следующими свойствами:

* код записи;
* ссылка на пользователя;
* ссылка на лекарство;
* дата начала приёма;
* дата окончания приёма;
* количество таблеток за приём;
* количество приёмов в день;
* общее количество таблеток за курс.

Категория полностью определяется названием.

Лекарство ссылается на сущность Категория.

Сущность Список приёма лекарств имеет следующие свойства:

* код приёма;
* ссылка на пользователя;
* ссылка на лекарство;
* дата принятия лекарства;
* количество принятых таблеток;

Для обеспечения первой нормальной формы информационной модели припишем каждому объекту уникальное ключевое поле (например, КодПользователя).

ER-диаграмма информационной модели предметной области приведена на рисунке А.1.

…………………………………………….фывфывфыв

Каждое событие превратится в новость. Но для разных сочетаний категории персонажа и категории мероприятия текст новости должен формироваться по разному. Например: "Иван Иванов (спортсмен) на чемпионате мира по футболу занял второе место", "Джон Джонсон (политик) присутствовал на чемпионате мира по футболу и заявил нашему корреспонденту, что ему всё понравилось". Потребуется список типов событий, представляющих собой текст с разметкой фасетов (мест, куда подставляются конкретные значения, например фасет для имени персонажа).

Возможна новость о том, как два или более персонажа участвуют в одном мероприятии. В информационной модели, описанной выше это означает наличие двух или более событий, ссылающихся на одно и то же мероприятие. Таких сочетаний может оказаться чрезвычайно много, в том числе слишком сложных для корректной формулировки новости.

Проще реализовать генерацию новостей в обратном порядке: выбрать тип события и подобрать необходимых для него персонажей и мероприятие. В этом случае типы событий могут отличаться друг от друга не только по категориям персонажа и мероприятия, но и по другим их свойствам (например, в своей или чужой стране для персонажа происходят события).

В этом случае единственный выход от хранения списка событий была бы возможность проследить, что один персонаж не участвует сразу в нескольких мероприятиях. Но этого же можно добиться, используя при случайном выборе алгоритм выбора без возврата. Таким образом, таблицу событий можно не реализовывать. Предполагается, что список категорий для персонажа и мероприятий единый и фиксированный с возможностью добавления. Это может быть реализовано как отдельный объект категория с единственным атрибутом название, являющимся ключевым полем.

Чтобы одно и то же по названию мероприятие не встречалось в лентах новостей слишком часто (чемпионат мира по футболу), мы можем трактовать объект мероприятие в более общем понимании, то есть скорее как тип мероприятия (соревнования, конференция и так далее), а конкретность ему будут придавать категория и место (соревнования по фигурному катанию в городе Берлин).

2.2 Описание программы

2.2.1 Общие сведения

.

2.2.2 Функциональное назначение

.

2.2.3 Описание логической структуры

2.2.3.1 Алгоритм программы

.

2.2.3.2 Используемые методы

Описанную в разделе 2.1 информационную модель можно реализовать разными методами: база данных (Access, MySQL и так далее), электронная таблица, текстовые файлы (.txt, .csv и другие) и константные массивы.

При реализации программы на языке программирования Python

2.2.3.3 Составные части программы и связи между ними

.

2.2.4 Используемые технические средства

.

2.2.5 Вызов и загрузка

.

2.2.6 Входные данные

.

2.2.7 Выходные данные

.

2.3 Руководство оператора

.

2.3.1 Назначение программы

.

2.3.2 Условия выполнения программы

.

2.3.3 Выполнение программы

.

Заключение

В ходе дипломного проектирования разработан программный продукт, соответствующий требованиям индивидуального задания. Это ...ПРОДОЛЖИТЬ...

Работы, выполненные в ходе проектирования, и разработанный программный продукт описаны в настоящей пояснительной записке с соблюдением требований ГОСТ ЕСПД.

Таким образом, задание на выпускную квалификационную работу выполнено в полном объёме.

Список литературы

6 Методические указания по оформлению курсовых и выпускной квалификационной работ: учеб. пособие для студентов ПТК НовГУ/ Авт.сост. Чернега А. М., ПТК НовГу им. Ярослава Мудрого. — Великий Новгород, 2006 – 27 с.

7 Положение об организации выполнения и защиты курсовой работы (проекта) в колледжах НовГУ [Электронный ресурс] URL: http://www.novsu.ru/file/990719 (Дата обращения 28.06.2014).

Приложение А

(обязательное)

Рисунок А.1 — ER-диаграмма информационной модели предметной области

Приложение Б

(обязательное)

Б.1 Текст программы «»

unit Unit1;

Приложение В

(обязательное)

Рисунок В.1 — Главное окно

Приложение Г

(обязательное)

#### Приложение В

продолжение

Рисунок Г.1 — Схема процедуры начальной подготовки FormCreate

Рисунок Г.2 — Схема взаимодействия процедуры начальной подготовки FormCreate

Приложение Д

(обязательное)

Рисунок Д.1 — Исполняемый