

Автоматизация работы врача терапевта в городской поликлинике на основе спиралевидной модели внедрения информационных систем (Часть 2)

Дудкин Дмитрий Игоревич

Аннотация: в статье проводится детальный анализ спиралевидной методологии внедрения систем, определяются требования и формулируется список требований, проектируются процессы и оргструктур в моделях AS-IS и TO-BE нотаций ARIS VACD и UML Activity Diagram. Для каждого витка в спиралевидной модели внедрения реализуются ключевые бизнес-процессы в среде MS Access, ведется тестирование и дается количественная оценка результатов испытания. Разработанная система позволяет упорядочить и систематизировать работу врача-терапевта. А это значительно повышает качество работы, ускоряет продолжительность приёма пациентов и сокращает очереди на прием к врачу.

2.4. Архитектура данных разрабатываемой системы

Используя процессы и данные выше, всю информацию можно разбить на классы, приведённые в таблице 2.1.

Таблица 2.1 - Классы данных

Название класса	Данные	Тип данных	Размерность
Личные данные пациентов	ФИО	Текстовые	65
	Дата рождения	Дата/время	Краткий формат даты
	Домашний адрес	Текстовые	70
	Номер паспорта	Текстовые	30
	Телефон	Текстовые	20
	Пол	Текстовые	1
	ОМС	Текстовые	20
Даты посещения	ФИО врача	Текстовые	85
	Пациент	Числовой	Длинное целое
	Дата посещения	Дата/время	Краткий формат даты
Причина посещения	Пациент	Числовой	Длинное целое
	Причина посещения	Текстовые	200
	Дата посещения	Числовой	Длинное целое

Название класса	Данные	Тип данных	Размерность
Лечение пациента	Пациент	Числовой	Длинное целое
	Причина посещения	Числовой	Длинное целое
	Дата посещения	Числовой	Длинное целое
	Лечение	Текстовые	255
Направление на анализ	Пациент	Числовой	Длинное целое
	ФИО врача	Числовой	Длинное целое
	Дата посещения	Числовой	Длинное целое
	Вид анализа	Текстовые	50
	Печать	Текстовые	1
Анализ крови	Пациент	Числовой	Длинное целое
	Показатели	Текстовые	50
	Норма	Текстовые	50
	Результаты анализа	Текстовые	50
	Вердикт	Текстовые	50
Анализ мочи	Пациент	Числовой	Длинное целое
	Показатели	Текстовые	50
	Норма	Текстовые	50
	Результаты анализа	Текстовые	50
	Вердикт	Текстовые	50
Анализ кала	Пациент	Числовой	Длинное целое
	Показатели	Текстовые	50
	Норма	Текстовые	50
	Результаты анализа	Текстовые	50
	Вердикт	Текстовые	50
Реабилитация пациента	Пациент	Числовой	Длинное целое
	Врач	Числовой	Длинное целое
	Дата посещения	Числовой	Длинное целое
	Дата проведения осмотра	Дата/время	Краткий формат даты
	Результат проведения осмотра	Текстовые	150
	Примечание	Текстовые	150

Проведем процедуру нормализации данных [6]. Стандартно все данные приводят к третьей нормальной форме (НФ), нормализации такого уровня в проектах внедрения информационных систем бывает достаточно. После разделения всех данных на классы и проведения процедуры нормализации, получается нормализованная таблица с

классами данных (табл.2.2) и архитектура данных разрабатываемого приложения, приведённая на рисунке 2.8.

Таблица 2.2 - Классы данных (нормализованная)

Название класса	Данные	Тип данных	Размерность
Личные данные пациентов	🔑 Код пациента	Счетчик	Длинное целое
	Фамилия	Текстовые	20
	Имя	Текстовый	15
	Отчество	Текстовый	30
	Дата рождения	Дата/время	Краткий формат даты
	Домашний адрес	Текстовые	70
	Номер паспорта	Текстовые	30
	Телефон	Текстовые	20
	Пол	Текстовые	1
ОМС	Текстовые	20	
Даты посещений	🔑 Код посещения	Счетчик	Длинное целое
	Фамилия врача	Текстовые	50
	Имя врача	Текстовый	15
	Отчество врача	Текстовый	20
	Код пациент	Числовой	Длинное целое
	Дата посещения	Дата/время	Краткий формат даты
Причина посещения	🔑 Код причины посещения	Счетчик	Длинное целое
	Код пациент	Числовой	Длинное целое
	Причина посещения	Текстовые	200
	Дата посещения	Числовой	Длинное целое
Лечение пациента	🔑 Код лечения	Счетчик	Длинное целое
	Код пациент	Числовой	Длинное целое
	Причина посещения	Числовой	Длинное целое
	Дата посещения	Числовой	Длинное целое
	Лечение	Текстовые	255
Направление на анализ	🔑 Код анализа	Счетчик	Длинное целое
	Код пациент	Числовой	Длинное целое
	Фамилия врача	Числовой	Длинное целое
	Имя врача	Числовой	Длинное целое
	Отчество врача	Числовой	Длинное целое
	Дата посещения	Числовой	Длинное целое

Название класса	Данные	Тип данных	Размерность
	Вид анализа	Текстовые	50
	Печать	Текстовые	1
Анализ крови	🔑 Код анализа	Счетчик	Длинное целое
	Код пациент	Числовой	Длинное целое
	Показатели	Текстовые	50
	Норма	Текстовые	50
	Результаты анализа	Текстовые	50
	Вердикт	Текстовые	50
Анализ мочи	🔑 Код анализа	Счетчик	Длинное целое
	Код пациент	Числовой	Длинное целое
	Показатели	Текстовые	50
	Норма	Текстовые	50
	Результаты анализа	Текстовые	50
	Вердикт	Текстовые	50
Анализ кала	🔑 Код анализа	Счетчик	Длинное целое
	Код пациент	Числовой	Длинное целое
	Показатели	Текстовые	50
	Норма	Текстовые	50
	Результаты анализа	Текстовые	50
	Вердикт	Текстовые	50
Реабилитация пациента	🔑 Код реабилитации	Счетчик	Длинное целое
	Код пациент	Числовой	Длинное целое
	Врач	Числовой	Длинное целое
	Дата посещения	Числовой	Длинное целое
	Дата проведения осмотра	Дата/время	Краткий формат даты
	Результат проведения осмотра	Текстовые	150
	Примечание	Текстовые	150

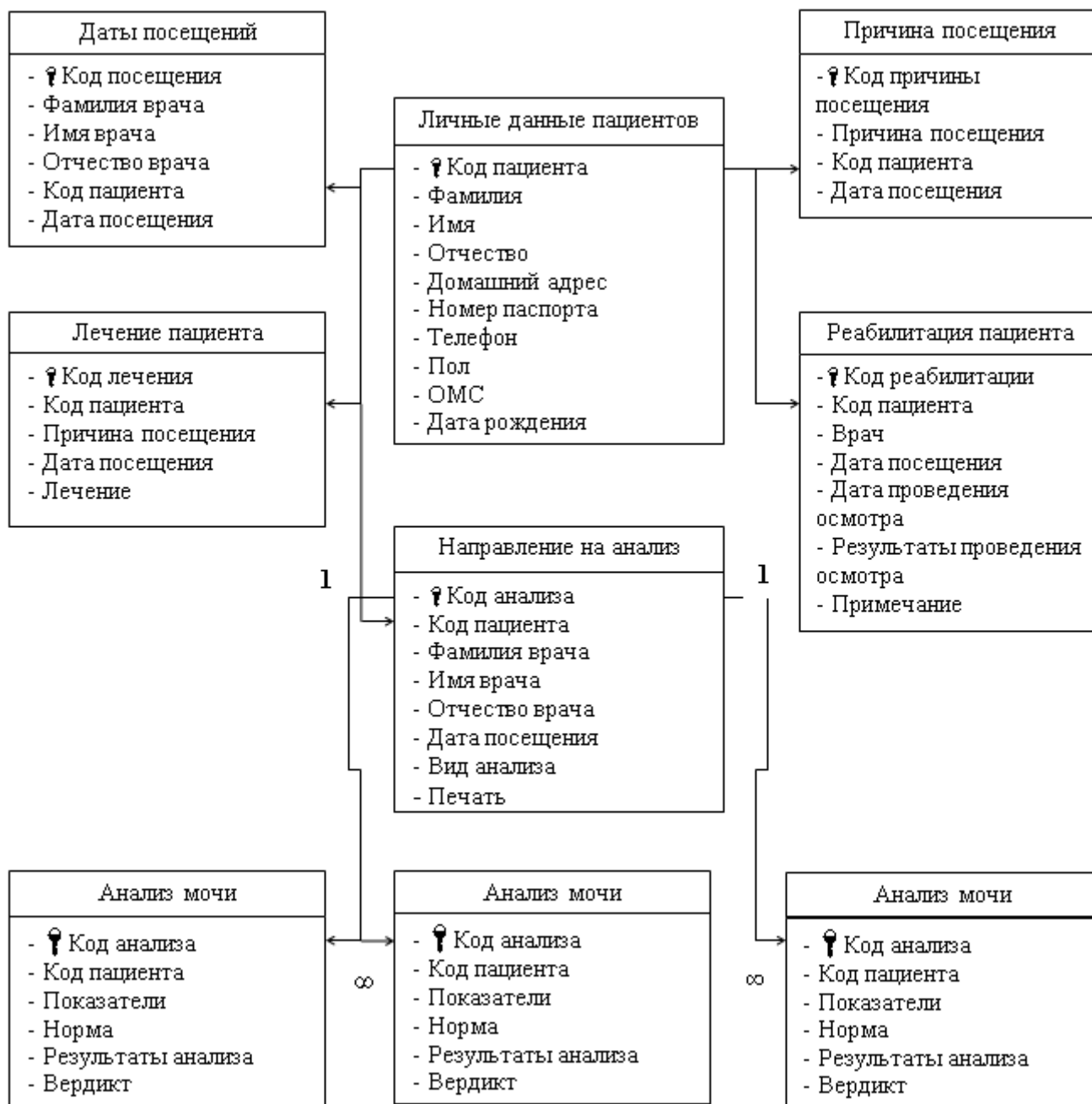


Рис. 2.8. - Архитектура данных разрабатываемого приложения

2.5. Описание разрабатываемого приложения

Для наиболее удобной работы конечного пользователя с программой необходимо создать предварительные макеты пользовательских интерфейсов программы. При входе в базу данных пользователю предстоит выбрать, нужно ли ему добавить нового

или посмотреть личные данные пациента, либо посмотреть историю его болезни. Главный интерфейс представлен на рисунке 2.9.

Врач терапевт

Добавление и поиск пациента		
Добавление даты посещения		
Добавление причины посещения		
Добавить лечение пациента		
Направление на анализ		
Анализ крови	Анализ мочи	Анализ кала
Реабилитация пациента		
Выход		

Рис. 2.9. - Интерфейс базы данных

Допустим, пользователь решил занести данные о новом пациенте, либо посмотреть информацию о пациенте. Для этого следует нажать на кнопку «Добавление и поиск пациента». На экране откроется окно с информацией о пациенте, которое представлено на рисунке 2.10.

Информация о пациенте

Код пациента	<input type="text"/>
Фамилия	<input type="text"/>
Имя	<input type="text"/>
Отчество	<input type="text"/>
Пол	<input type="text"/>
Дата рождения	<input type="text"/>
Домашний адрес	<input type="text"/>
Номер паспорта	<input type="text"/>
Телефон	<input type="text"/>
ОМС	<input type="text"/>

Рис. 2.10. - Интерфейс «Добавление и поиск пациента»

Далее необходимо добавить дату посещения. Для этого следует «Добавление даты посещения». На экране появится окно, представленное на рисунке 2.11. После внесения нужной информации, она отобразится в таблице «Дата посещений».

Добавление даты посещения

Данные пациента	<input type="text"/>
Фамилия врача	<input type="text"/>
Имя врача	<input type="text"/>
Отчество врача	<input type="text"/>
Дата посещения	<input type="text"/>

Рис. 2.11. - Интерфейс «Добавление даты посещения»

Если пользователю необходимо добавить историю болезни пациента, либо ее посмотреть, следует нажать на кнопку «Добавление причины посещения». На экране появится окно с историей болезни соответствующего пациента, представленное на рисунке 2.12. В него вносятся такие данные, как фамилия, имя, отчество пациента, причина посещения, дата посещения. После внесения нужной информации она отобразится в таблице «Причина посещения».

Добавление причины посещения

Данные пациента	<input type="text"/>
Причина посещения	<input type="text"/>
Дата посещения	<input type="text"/>

Рис. 2.12. - Интерфейс «Добавление причины посещения»

После следует внести информации о лекарствах, которые назначил врач пациенту. Для этого потребуется нажать на кнопку «Добавить лечение пациента». На экране отобразится окно, данное на рисунке 2.13. После внесения нужной информации она появится в таблице «Лечение пациента».

Лечение пациента

Пациента	
Дата посещения	
Причина посещения	
Лечение	

Рис. 2.13. - Интерфейс «Добавить лечение пациента»

Если врач не может точно определить, в чём заключается причина болезни, то пациент направляется на сдачу анализов. Для этого врач-терапевт должен распечатать пациенту направление на анализ. Форма для печати представлена на рисунке 2.14.

Направление на анализ

Пациент	
Фамилия врача	
Имя врача	
Отчество врача	
Дата посещения	
Вид анализа	

Рис. 2.14. - Интерфейс «Направление на анализ»

После того, как будут готовы результаты анализы пациента, врач может назначить лечение. Результаты анализа будут заноситься в форму, представленную на рисунке 2.15. Данная форма представлена для анализа крови. Формы для анализа мочи и кала будет иметь схожий вид.

Результаты анализа крови

Пациент	
Фамилия врача	
Имя врача	
Отчество врача	
Дата посещения	

Показатели	Норма	Результаты	Вердикт

Рис. 2.15. - Интерфейс «Анализ крови»

В случае плохого самочувствия пациента, появления осложнений или плохих анализов, пациента следует положить в палату для наблюдения. Результаты осмотров будут заноситься в форму, представленную на рисунке 2.16. В примечание будет заноситься информация о состоянии пациента.

Реабилитация пациента

Пациент	
Врач	
Дата посещения	
Дата проведения осмотра	
Примечание	

Рис. 2.16. - Интерфейс «Реабилитация пациента»

Работа программы будет вестись согласно представленной схеме приложения, указанное на рисунке 2.17.

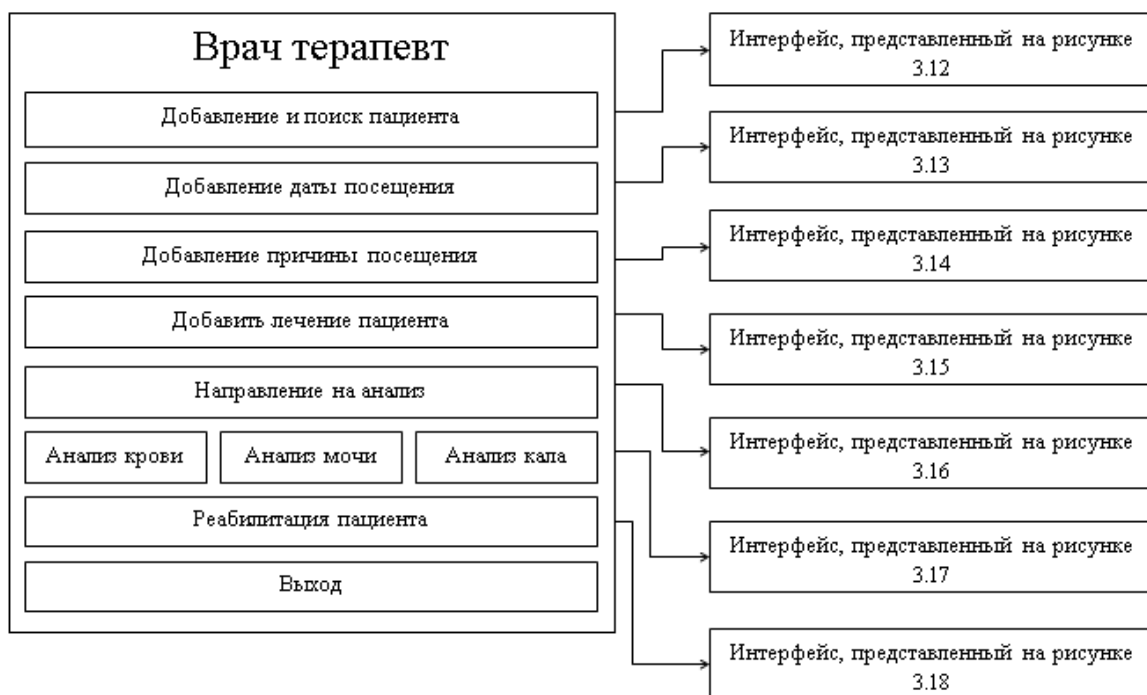


Рис. 2.17. - Схема программы

3. Разработка приложения

3.1. Первый прототип программы (I виток спирали)

Реализация программы на первом витке спирали представляет собой заведение таблиц с элементами управления самой среды разработки MS Access. Здесь нет никакого интерфейса, который мог бы облегчить работу пользователя с программой.

Основным риском на данном этапе разработки программы являются возникновение проблем с электропитанием. При возникновении данной проблемы возможен сбой работы программы. Это может привести к отказу работы базы данных, а также не исключается возможность потери каких-либо данных. Для решения данной проблемы требуется иметь дополнительный источник электропитания/электричества. На рисунках 3.1-3.2 представлены фрагменты таблиц «Личные данные пациента» и «Дата посещения», созданные в программе MS Access.

Личные данные пациентов

Код пациента	Фамилия	Имя	Отчество	Дата рождения	Домашний адрес	Номер паспорта	Телефон	Пол	ОМС
1	Сидоров	Владимир	Петрович	13.03.1986	ул.Победы,д.8,кв.24	1324 465234	8-(916)-723-64-37	М	3244 3645 6457 6457
2	Иванов	Сергей	Владимирович	23.08.1991	ул.Севастопольская,д.12,кв.57	1234 748924	8-(903)-235-73-33	М	2323 8597 2983 5758
3	Кастров	Дмитрий	Игоревич	11.02.1989	ул.Проспект мира,д.20,кв.28	1426 274365	8-(911)-136-23-56	М	3453 4534 9059 3458
4	Васильев	Олег	Васильевич	24.08.1994	ул.Ленина,д.3,кв.5	1846 192385	8-(952)-375-92-74	М	3450 9345 9834 8505
5	Ларин	Иван	Сергеевич	05.01.1988	ул.Шлыково,д.31,кв.67	1385 283785	8-(903)-284-75-83	М	6579 8324 9234 8659

Рис. 3.1. - Фрагмент таблицы «Личные данные пациента» базы данных

Даты посещения

Код посещения	Фамилия врача	Имя врача	Отчество врача	код пациента	Дата посещения
4	Паршина	Елена	Викторовна	Сидоров	08.02.2018
5	Паршина	Елена	Викторовна	Ларин	08.03.2018
6	Паршина	Елена	Викторовна	Кастров	16.03.2018
7	Паршина	Елена	Викторовна	Васильев	30.03.2018

Рис. 3.2. - Фрагмент таблицы «Дата посещения» базы данных

После того, как все таблицы были созданы, мы создаем схему и связи всех используемых таблиц данных, необходимых для приложения. Схема данных, представленная на рисунке 3.3, строилась путем добавления таблиц данных, выделением из них ключевых полей, необходимых для создания связей между ними. Все это выполнялось в режиме «конструктора».

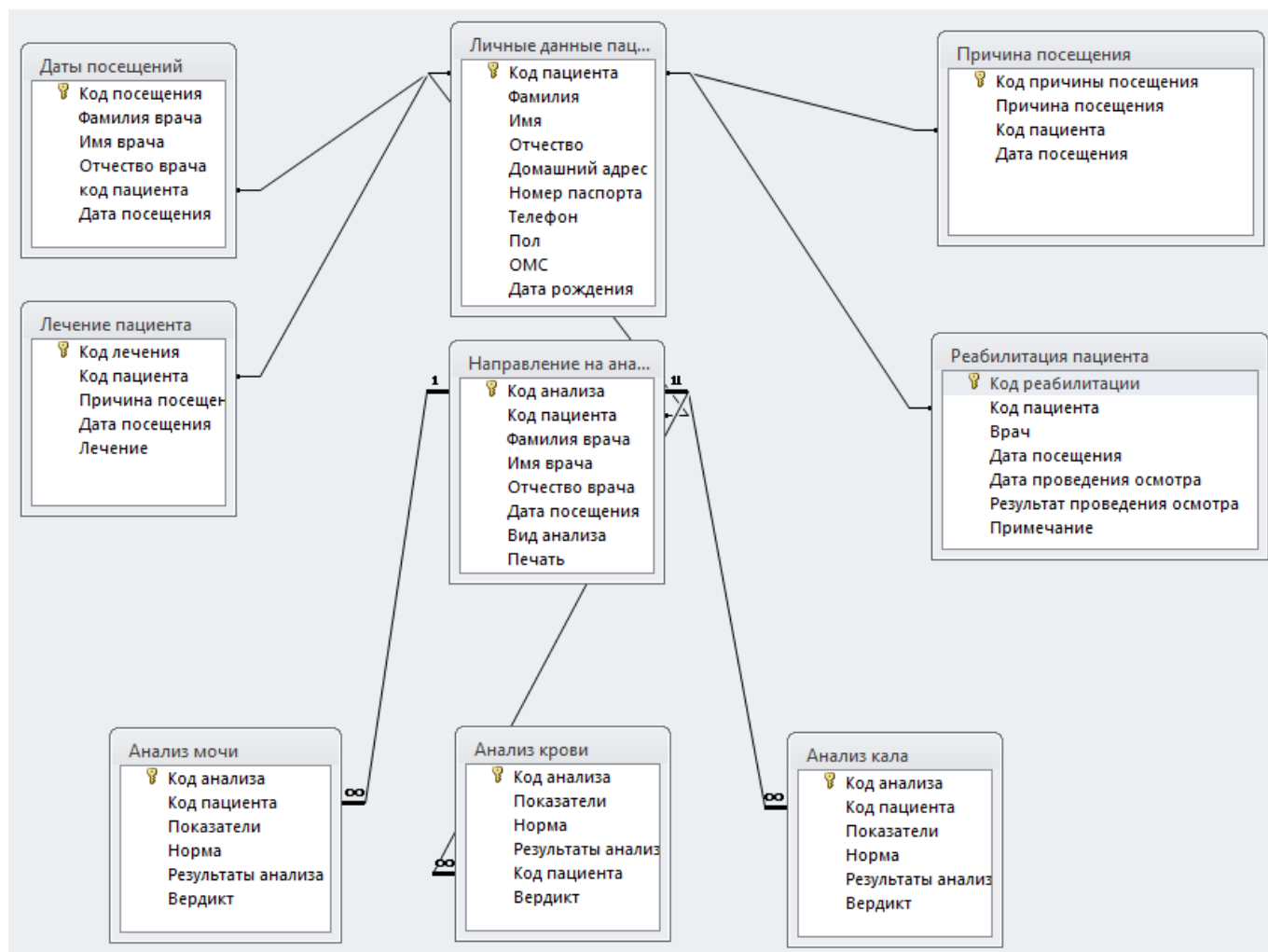


Рис. 3.3. - Схема данных

3.2. Второй прототип программы (II виток спирали)

На данном этапе реализуется возможность просмотра данных, а также возможность изменения, удаления, внесения и поиска информации о пациенте, а также для печати (рисунки 3.4-3.5). Для чего создаётся «Форма» - объект, с помощью которого пользователи могут добавлять, редактировать и отображать данные.

Основным риском на данном этапе разработки программы является человеческий фактор-невнимательность врача при внесении данных пациента. Со стороны пользователя ПО: врача, необходимо быть более внимательным при работе с персональными данными пациентов.

Информация о пациенте

Код пациента	<input type="text" value="1"/>	Добавить пациента
Фамилия	<input type="text" value="Сидоров"/>	
Имя	<input type="text" value="Владимир"/>	Поиск пациента
Отчество	<input type="text" value="Петрович"/>	Удалить пациента
Пол	<input type="text" value="М"/>	Сохранить изменения
Дата рождения	<input type="text" value="13.03.1986"/>	
Домашний адрес	<input type="text" value="ул.Победы,д.8,кв.24"/>	Печать
Номер паспорта	<input type="text" value="1324 465234"/>	
Телефон	<input type="text" value="8-(916)-723-64-37"/>	Вернуться на главную страницу
ОМС	<input type="text" value="3244 3645 6457 6457"/>	

Рис. 3.4. - Форма «Добавление пациента» с возможностью поиска, удаления, добавления пациентов и печати информации

Добавление даты посещения

Данные пациента	<input type="text" value="Сидоров"/>	Поиск пациента
Фамилия врача	<input type="text" value="Паршина"/>	Сохранить
Имя врача	<input type="text" value="Елена"/>	Добавить
Отчество врача	<input type="text" value="Викторовна"/>	Удалить
Дата посещения	<input type="text" value="08.02.2018"/>	Выход на главную страницу

Рис. 3.5. - Форма «Добавление даты посещения» с возможностью поиска, добавления и удаления даты посещения пациента

3.3. Конечный продукт (III виток спирали)

Следующим шагом требовалось защитить данные пациентов. Для этого следовало поставить пароль на базу данных (рисунок 3.6). Основным риском на данном этапе разработки программы является введение слишком легкого/простого пароля, а также небрежного отношения со стороны пользователей за соблюдением условий работы в ПО. Это может привести к входу в базу данных посторонних лиц. Для решения данной проблемы требуется тщательно подходить к выбору пароля, а врачу-пользователю

внимательно подходить к хранению и использованию служебной информации (пароли, коды) и персональных данных пациентов.

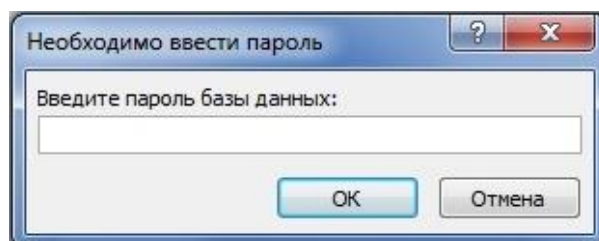


Рис. 3.6. - Окно для ввода пароля

Так же следует создать начальный интерфейс для ускорения приёма пациентов и упрощения работы пользователя (рисунок 3.7).

Врач терапевт

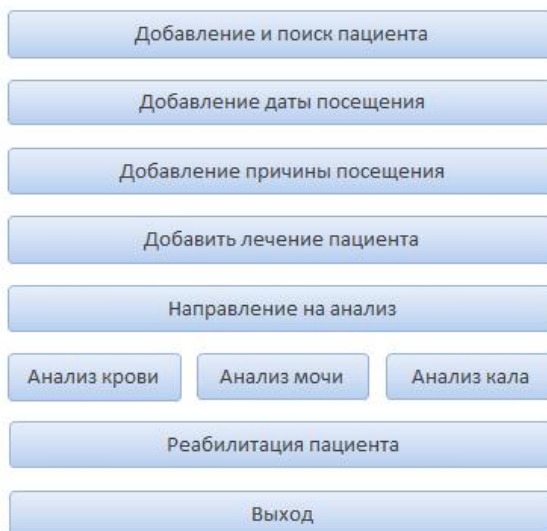


Рис. 3.7. - Главное меню приложения

При нажатии на кнопку «Добавление и поиск пациента» происходит переход в форму с информацией о пациентах. В ней можно произвести поиск информации о

нужном нам пациенте, добавить нового при нажатии «Добавить пациента», удалить пациента, либо отредактировать данные, которые уже существуют в этой форме.

Вторая кнопка - «Добавление даты посещения», при ее нажатии происходит переход в форму с датами посещений пациентов. В ней можно посмотреть даты посещений пациентов и добавить новые. Кнопка «Добавление причины посещения» позволяет перейти в форму с причинами посещений всех пациентов. В ней можно посмотреть историю болезней пациентов, а также добавить новые.

Опция «Добавить лечение пациента» обеспечивает переход в форму со всей информацией об истории лечения пациентов. В ней можно посмотреть историю лечения пациента, а также добавить новое, в случае необходимости. Далее кнопка «Направление на анализ» позволяет сформировать направление пациенту на какой-либо анализ и распечатать его.

Кнопка «Анализ крови/мочи/кала» показывает форму с результатами анализов пациентов. В ней можно посмотреть информацию о результатах анализов для принятия решения о том, чем производить лечение пациента. Девятая кнопка «Реабилитация пациента» отображает информацию о том, как происходит реабилитация пациента. В случае осложнения/ухудшения состояния назначить новые лекарства.

4. Тестирование разработанного приложения

Тестирование программного обеспечения - процесс анализа программного продукта и сопутствующей документации с целью выявления недостатков в работе и повышения его качества [7]. Обычно для проверки работоспособности разработанной программы проводят три вида тестирования: функциональное, интеграционное и нагрузочное. Акцентируем внимание на последнем виде испытаний.

Нагрузочное тестирование проводилось для исследования быстроты действия таких процессов, как, запись пациента на прием (поскольку этот процесс призван уменьшить время ожидания пациентов в очередях, то предполагается, что его выполнение не должно занимать много времени) и поиск записи. Для тестирования было выбрано следующее количество записей: 5 (в среднем посетителей за час работы), 50 (в среднем посетителей за день) и 500 (в среднем посетителей на 2 недели). Результаты нагрузочного тестирования приведены в табл.4.1.

Вначале 5 раз проводится проверка отклика системы на различные действия, ее результаты вносятся в соответствующие строки столбцов t1-t5. Далее рассчитывается среднее арифметическое всех измерений по формуле (1).

$$t_{\text{ср.ариф.}} = \frac{\sum_{i=1}^n t_i}{n} \quad (1)$$

Результат заносится в столбец «Среднее время отклика» таблицы ниже. Далее находятся соответствующие средние квадратические отклонения по формуле (2):

$$\sigma = \sqrt{\frac{1}{n} * \sum_{i=1}^n (t_i - t_{\text{ср.ариф.}})^2} \quad (2)$$

Погрешность измерений рассчитывалась по формуле (3), где n - число измерений, - доверительный коэффициент Стьюдента, равный 0.95, - абсолютная погрешность электронного секундомера равная 0,005.

$$\Delta t = \sqrt{\left(\frac{\sigma}{n} \cdot t_{\alpha(N-1)}\right)^2 + \Delta t_p^2} \quad (3)$$

В столбце «Время отклика» дано итоговое время отклика согласно формуле (4):

$$t_{\text{откл}} = t_{\text{ср.ариф.}} \pm \Delta t \quad (4)$$

Таблица 4.1 - Результаты нагрузочного тестирования

Количество записей	Действие	t1, с.	t2, с.	t3, с.	t4, с.	t5, с.	Среднее время отклика, с.	Средн. квадр. отклон., с.	Погрешность измерений, с.	Время отклика, с.
5	Добавление	0,21	0,23	0,21	0,23	0,2	0,216	0,0235	0,0254	0,216±0,025
	Поиск	0,2	0,22	0,19	0,21	0,2	0,206	0,0225	0,0243	0,206±0,024
50	Добавление	0,3	0,29	0,31	0,31	0,3	0,302	0,0341	0,0369	0,302±0,037
	Поиск	0,27	0,26	0,28	0,29	0,3	0,28	0,0339	0,0358	0,28±0,036
500	Добавление	0,39	0,38	0,39	0,37	0,4	0,385	0,0376	0,0394	0,385±0,04
	Поиск	0,36	0,35	0,36	0,34	0,4	0,362	0,0369	0,0389	0,362±0,039

Как видно из табл. 4.1, время отклика не превышает 0,5 секунд и почти не замечается пользователем. Таким образом, можно сделать вывод, что нагрузочное тестирование проведено успешно.

5. Заключение

Целью данной работы являлось создание автоматизированного рабочего места для врача-терапевта. Необходимо было оптимизировать рабочий процесс с целью экономии времени сотрудника и уменьшение времени приема пациентов. Для чего были выполнены следующие упражнения:

- проанализированы пользовательские требования (19 требований), полученные путем опроса и изучения документации. Определены приоритеты полученных требований;
- составлены функциональные требования, которые должны были быть реализованы в разрабатываемом приложении;
- список требований разделен на 1-3 приоритета разработки;
- составлены модели AS-IS и TO-для BE ключевых бизнес-процессов организации в нотациях ARIS VACD (0-1 уровни) и UML Activity Diagram (2-3 уровень);
- проведена разработка в среде MS Access, где поэтапно были реализованы все требования, исправлены недочеты и ошибки в работе программы;
- в финале было проведено нагрузочное тестирование для различного количества записей в БД, результаты которого свидетельствовали о высокой производительности программы.

Литература

1. А.В. Варзунов, Е.К. Торосян, Л.П. Сажнева. "Анализ и управление бизнес-процессами". Учебное пособие, 2010 - 212 с.
2. И.В. Абрамов. Методические указания по дисциплине "Модели и методы информационно-управляющих систем". Ижевск, 2004 - 314 с.
3. Владимир Репин, Виталий Елиферов. "Процессный подход к управлению". Моделирование бизнес-процессов. Издательство "Манн, Иванов и Фербер", Москва, 2013 - 215 с.
4. Ю.А. Ларина. "Основы объектно-ориентированного моделирования с использованием языка UML". Учебное пособие, Ярославль, 2010 - 152 с.
5. Ю.А. Ларина. "Основы объектно-ориентированного моделирования с использованием языка UML". Учебное пособие, Ярославль, 2010 - 152 с.
6. Дейт К. Дж. Введение в системы баз данных / пер. с англ. и ред. К. А. Птицына - 8-е изд. - М.: Вильямс - 2016. - 327 с.

7. Штенников Д.Г. Разработка информационных систем в образовании. Учебное пособие. - СПб: СПбГУ ИТМО, 2012. - 242 с.

Выходные данные статьи

Дудкин Д.И. Автоматизация работы врача терапевта в городской поликлинике на основе спиралевидной модели внедрения информационных систем (Часть 2) // Корпоративные информационные системы. - 2022. - №1 (17) – С. 1-18. – URL: <https://corpinfosys.ru/archive/issue-17/183-2021-17-therapistautomation>.

Об авторе



Дудкин Дмитрий Игоревич - студент 4-го курса кафедры оптических и биотехнических систем и технологий физико-технологического института РТУ МИРЭА. Тема выпускной квалификационной работы бакалавра «Автоматизация ключевого бизнес-процесса врача терапевта в городской поликлинике на основе спиралевидной модели внедрения информационных систем». Электронная почта: mail@corpinfosys.ru.