

Применение спиралевидной модели внедрения информационных систем для роботизации городской больницы на основе программных решений MS Access и UiPath

Чепров Евгений Дмитриевич

Аннотация: в статье демонстрируется применение спиралевидной модели внедрения информационных систем для автоматизации работы городской больницы. Следуя спиралевидной модели внедрения, проект разбит на 4-е цикла разработок, по результатам которых получается промежуточный рабочий продукт. Приложение для работы медицинского учреждения реализуется в среде MS Access, ключевые бизнес-процессы роботизируются с использованием программного продукта UiPath.

Введение

Одной из проблем работы лечебных учреждений, которые определяются качеством и своевременностью оказания медицинских услуг, является сложность обработки информации о пациентах и персонале, в частности:

- длительный и трудоемкий процесс поиска необходимых сведений о пациентах в медицинских картах;
- отсутствие или несоблюдение единого формата внесения данных о пациенте в карту, в запись поставленного диагноза, в отчет о проведенном лечении и др.;
- потеря бумажных носителей в процессе работы больницы.

Рост количества пациентов, обращающихся в лечебные учреждения за медицинской помощью, не должен снижать уровень и скорость предоставления услуг. Поэтому автоматизация отдельных бизнес-процессов функционирования лечебных учреждений видится целесообразной. Этого можно достичь за счет разработки приложения с помощью систем управления базами данных (СУБД), например, MS Access, и его применения в лечебных организациях. Программа позволит упростить ввод информации, сделает ее обработку более быстрой и удобной. Кроме того приложение на базе MS Access совместимо с операционной системой Microsoft Windows, которая используется на большинстве компьютеров медицинских

учреждений. Среду MS Access можно применять без наличия специальных знаний, навыков и подготовки [1].

Разработанную программу можно в дальнейшем автоматизировать, например, применяя средства роботизации бизнес-процессов UiPath [2]. Данное программное обеспечение позволяет роботизировать процессы любой сложности. Последнее избавляет человека от необходимости делать все вручную и передает часть задач по работе с данными на выполнение компьютеру. Целью текущей работы является демонстрация применения спиралевидной методологии внедрения медицинских информационных систем с использованием MS Access и UiPath на примере роботизации работы городской больницы.

1. Модели внедрения информационных систем

Информационные системы должны удовлетворять интересам бизнес пользователей, а также быть легко модифицируемыми и недорогими [3]. Плохо спроектированная и реализованная программная система потребует больших трудозатрат для ее поддержки и обновления. Одним из базовых понятий методологии проектирования ИС является жизненный цикл программного обеспечения (ЖЦ), определяющийся как процесс развития системы, продукта, услуги, проекта или других изготовленных человеком объектов, начиная со стадии концепции и заканчивая прекращением промышленного применения. Модель жизненного цикла – структура действий, связанных с жизненным циклом, организуемая в стадии для установления взаимосвязей и взаимопонимания всех вовлеченных сторон [4].

Существует множество различных моделей жизненного цикла ПО, одной из которых является спиральная модель [5]. В ней делается упор на начальные этапы ЖЦ системы: анализ и проектирование. Спиральная модель – классический пример применения эволюционной стратегии разработки программ. Особенность данной модели заключается в том, что программное обеспечение создается не сразу, а по частям с использованием метода прототипирования. Прототип представляет собой работоспособную программу, реализующую отдельную функцию или графический пользовательский интерфейс. Как показано на рис. 1.1, модель определяет четыре действия, каждое из которых соответствует квадранту спирали:

- подготовка – сбор требований и ограничений;
- планирование – формирование плана проекта и анализ рисков;

- моделирование и разработка - подготовка моделей и реализация программного продукта следующего уровня;
- развертывание - оценка заказчиком текущей версии продукта.



Рис. 1.1. Спиралевидная модель: 1 - начальный сбор требований проекта; 2 - та же работа, но на основе рекомендаций заказчика; 3 - планирование проекта и анализ риска с использованием начальных требований; 4 - планирование и анализ рисков; 5 - переход к комплексной системе; 6 - начальный макет системы; 7 - версия системы следующего уровня; 8 - разработанная система; 9 - оценивание заказчиком

Разработка здесь отображается движением по разворачивающейся спирали по часовой стрелке, причем старт проекта происходит в первом квадранте. С каждым витком спирали строятся все более полные версии ПО.

Спиральная модель позволяет начинать работу над следующим этапом, не дожидаясь завершения предыдущего. При необходимости на каждом цикле можно корректировать требования к разрабатываемому продукту. Главная проблема спиралевидной модели состоит в определении момента перехода на следующий этап. Возможным её решением является задание временных рамок для каждого цикла.

Отличительной особенностью этой модели служит детальное рассмотрение рисков, влияющих на организацию жизненного цикла. Барри Боэм, автор этой модели, в своей работе формулирует десять наиболее распространённых рисков, среди которых хочется выделить три наиболее значимые и актуальные [5]:

- нереалистичные сроки и бюджет;
- перфекционизм;
- непрерывный поток изменений.

Фундаментальным принципом реинжиниринга организации является рассмотрение деятельности компании не с точки зрения функционирования ее структурных подразделений, а с точки зрения организации и протекания в ней бизнес-процессов. Бизнес-процесс – есть связанное множество внутренних видов деятельности компании, заканчивающихся созданием продукции или услуги, необходимых потребителю. Под потребителем следует понимать как клиента, так и другой процесс, протекающий во внешнем окружении, например, у партнеров или субподрядчиков.

Бизнес-модель – это описание процессов, которое характеризует их основные элементы и связи между ними, и позволяет создать упрощенное представление операции с отражением наиболее существенных ее характеристик. Моделирование бизнес-процессов является ключевой составляющей проектов по реинжинирингу операций и внедрению крупномасштабных систем ПО. Отсутствие таких моделей служит одной из главных причин провала проектов [6].

Ориентирование на процессы является важным фактором успешного реинжиниринга деятельности компании. Другим не менее важным обстоятельством является переход предприятия к использованию новых цифровых технологий. Применение новейших технологий может привести не только к принципиальным изменениям в деятельности сотрудников, но и к полной замене существующих бизнес-процессов.

На начальных этапах создания программного решения необходимо четко понять, как функционирует организация. Руководитель хорошо знает работу в целом, но не в состоянии вникнуть в детали работы каждого рядового сотрудника. Сотрудники хорошо знают, что творится на их рабочих местах, но не соседних. Поэтому для проектирования работы предприятия необходимо построить адекватную бизнес-модель, которая будет содержать в себе знания всех участников бизнес-процессов организации.

При проектировании бизнес-процессов используют две модели: AS-IS (как есть) и TO-BE (как будет) [6]. Модель AS-IS основана на совокупности должностных инструкций, отчетов, приказов и нормативной документации предприятия. Она позволяет выяснить, «что мы делаем сегодня» перед тем, как перейти к тому, «что мы

будем делать завтра». Анализ модели позволяет выявить слабые места в работе организации, определить, в чем будут состоять преимущества усовершенствованных процессов и понять, насколько глубоким изменениям подвергнется существующая организация.

Найденные в модели недостатки исправляются в схеме ТО-ВЕ, представляющей собой новый способ функционирования предприятия. Модель ТО-ВЕ нужна для анализа альтернативных путей решения задач и выбора наилучших из них.

2. Анализ требований городской больницы

Для реализации программного решения необходимо собрать пользовательские требования к продукту. Этот этап является одним из важнейших в проекте. Без понимания требований заказчика невозможно создать адекватное программное обеспечение. Путем интервьюирования сотрудников клиники, были определены следующие требования:

- необходимость хранения информации о пациентах;
- возможность добавления, изменения и удаления информации;
- автоматическое сохранение изменений в таблицах баз данных.

Работа с требованиями создает необходимость формирования сводной таблицы с описанием всех пользовательских требований. Данная таблица позволит отслеживать требования и значительно облегчит дальнейшие шаги по созданию программного обеспечения для городской больницы [7]. Полный список требований представлен в табл. 2.1.

Таблица 2.1. - Список требований

Требования пользователя	Функциональное требование
Хранение информации о пациентах	Наличие таблицы с информацией о пациентах
Хранение информации о персонале	Наличие таблицы с информацией о персонале
Хранение информации о состоянии пациента	Наличие таблицы с информацией об анамнезе
Хранение информации об операциях	Наличие таблицы с информацией об операциях
Добавление записей	Наличие пользовательского

Требования пользователя	Функциональное требование
Удаление записей	интерфейса
Изменение записей	
Самостоятельный запуск БД	Наличие последовательностей для запуска БД и открытия разделов
Переход к нужному разделу	
Автоматический информации из других источников	Возможность открывать документ и копировать информацию оттуда
Автоматическое переключение между полями при изменении информации	Наличие автонажатия после изменения информации в ячейке
Автоматическое сохранение	Принудительное сохранение после любых действий с БД
Автоматический выход на главный экран	Наличие самозакрытия раздела после его использования
Автоматическое закрытие БД	Наличие последовательности для выхода из СУБД

Для выполнения проекта разобьем потребности на 4-е витка спирали, где с увеличением номера уменьшается приоритет требования и разработки соответственно. Это позволит создавать программу итеративно и соотносить каждый промежуточный результат с ожиданиями клиента. На первом витке будет добавлена функция хранения информации путем создания таблиц баз данных. Далее будут заданы связи между созданными таблицами. На следующем цикле разработки создается пользовательский интерфейс, который позволит обрабатывать информацию из таблиц баз данных. Создание роботизированных скриптов, определяющих пользовательские действия по вводу данных в программу, задаст третий виток разработки. Заключительный этап позволит связать роботизированные алгоритмы для ввода данных и приложение, при этом информация будет копироваться из внешних файлов.

3. Модели проектирования AS-IS и TO-BE

IDEFO – графическая нотация функционального моделирования, предназначенная для формализации и описания бизнес-процессов [8]. Отличительной особенностью IDEFO является её акцент на соподчинённость объектов. В IDEFO рассматриваются

логические отношения между работами, а не их временная последовательность в отличие от IDEF3.

Стандарт IDEF0 представляет организацию как набор модулей. Здесь существует правило: наиболее важная функция находится в верхнем левом углу диаграммы. Описание выглядит как логическая последовательность операций с входами, выходами, управляющими элементами и механизмами, которая постепенно детализируется до необходимого уровня точности. Чтобы быть правильно понятым, существуют правила использования графических нотаций модели IDEF0 (рис. 3.1).

Несмотря на то, что нотация IDEF0 имеет множество преимуществ, ее использование ограничено применением только на верхних уровнях декомпозиции процессов, где отсутствует чрезмерная детализация. Поэтому для реализации программного продукта необходимо использовать ещё одну нотацию моделирования процессов, в качестве которой будет применяться BPMN SLD.

Графический элемент	Описание
	Процесс
	Входящие данные процесса
	Исходящие данные процесса
	Ограничение процесса
	Ресурс процесса

Рис. 3.1. Графические элементы нотации IDEF0

Нотация SLD использует базовый набор интуитивно понятных графических элементов, которые позволяют описывать сложные семантические конструкции [9]. Кроме того спецификация SLD определяет, как диаграммы, описывающие бизнес-процесс, могут быть трансформированы в исполняемые модели. Основная цель SLD – создание стандартного набора условных обозначений понятных всем пользователям (рис. 3.2). Следовательно, SLD служит связующим звеном между фазой дизайна бизнес-процессов и фазой их программной реализации.

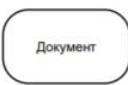
Графический элемент	Описание
	Ответственный организационный уровень
	Процесс
	Входящий/исходящий документ
	Условие

Рис. 3.2. Графические элементы нотации BPMN SLD

4. Проектирование процессов с помощью IDEF0 и BPMN SLD в модели AS-IS

Проектирование процессов в модели AS-IS позволяет показать функциональную схему медицинского учреждения в том виде, как она выглядит на данный момент. Функционирование поликлиники в нотации IDEF0 на нулевом уровне детализации состоит из трех бизнес-процессов: А1 - "Нанять сотрудника", А2 - "Положить пациента в стационар" и А3 - "Вылечить пациента". Эти процессы, а также связи между ними, показаны на рис. 4.1.

Ключевые бизнес-процессы на начальном уровне описания в модели AS-IS

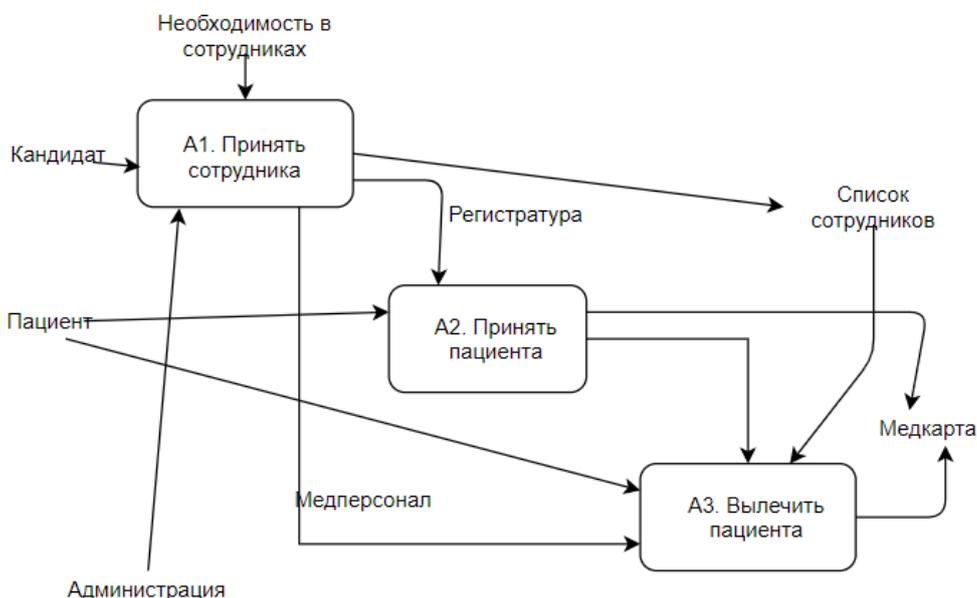


Рис. 4.1. Ключевые бизнес-процессы на начальном уровне описания в модели AS-IS

Подвергнем процессы A1 и A3 декомпозиции, тогда мы получим описание подпроцессов в нотации IDEF0 на 1-м уровне детализации для каждого из них (рис. 4.2-4.3).

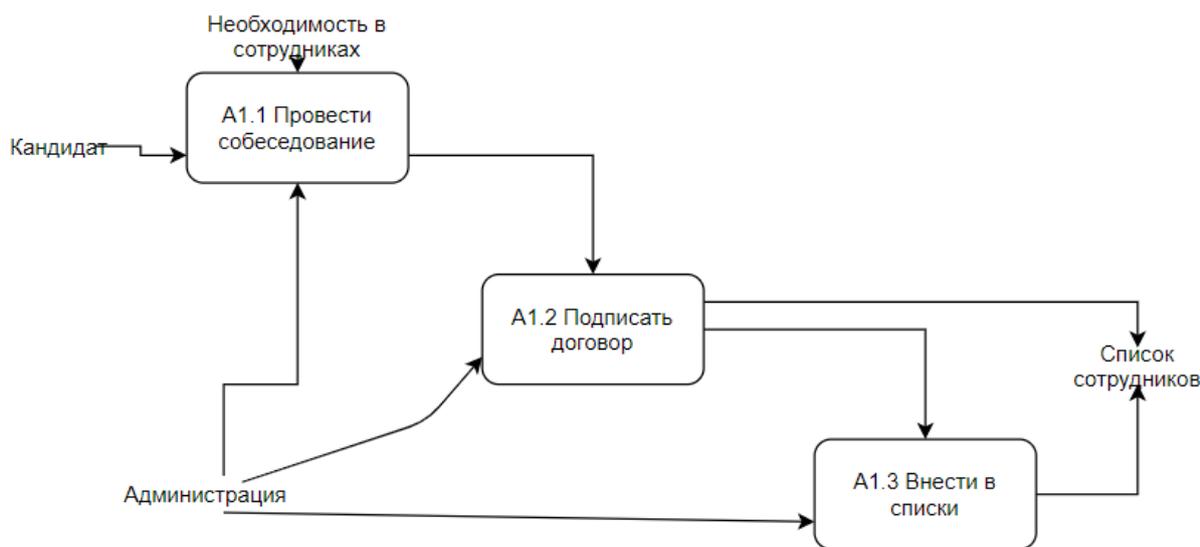


Рис. 4.2. Бизнес-процесс A1 - "Нанять сотрудника" на 1-м уровне детализации в модели AS-IS

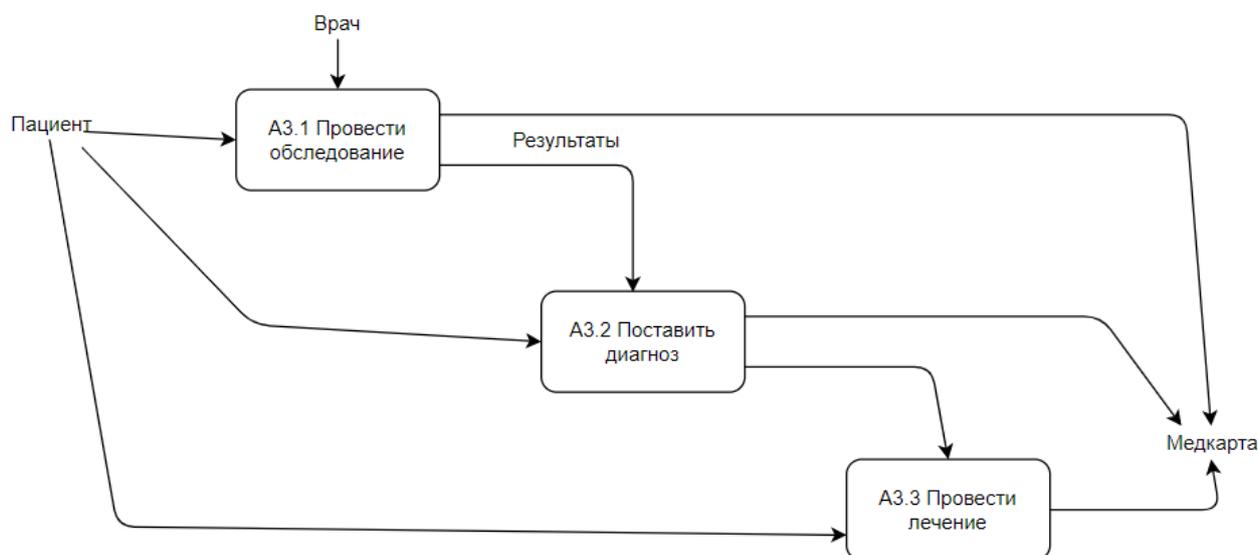


Рис. 4.3. Бизнес-процесс А3 - "Вылечить пациента" на 1-м уровне детализации в модели AS-IS

Подпроцесс А3.3 - "Провести лечение" нуждается в более подробном рассмотрении, поэтому подвергнем его декомпозиции. Однако моделирование на 2-м уровне с помощью IDEFO не наглядно, так как нотация используется только на верхних уровнях описания, исключая излишнюю детализацию. Тогда для моделирования хорошо подойдет нотация BPMN SLD, о которой писалось ранее. Результат декомпозиции подпроцесса лечения на 2-м уровне с помощью BPMN SLD дан на рис. 4.4.

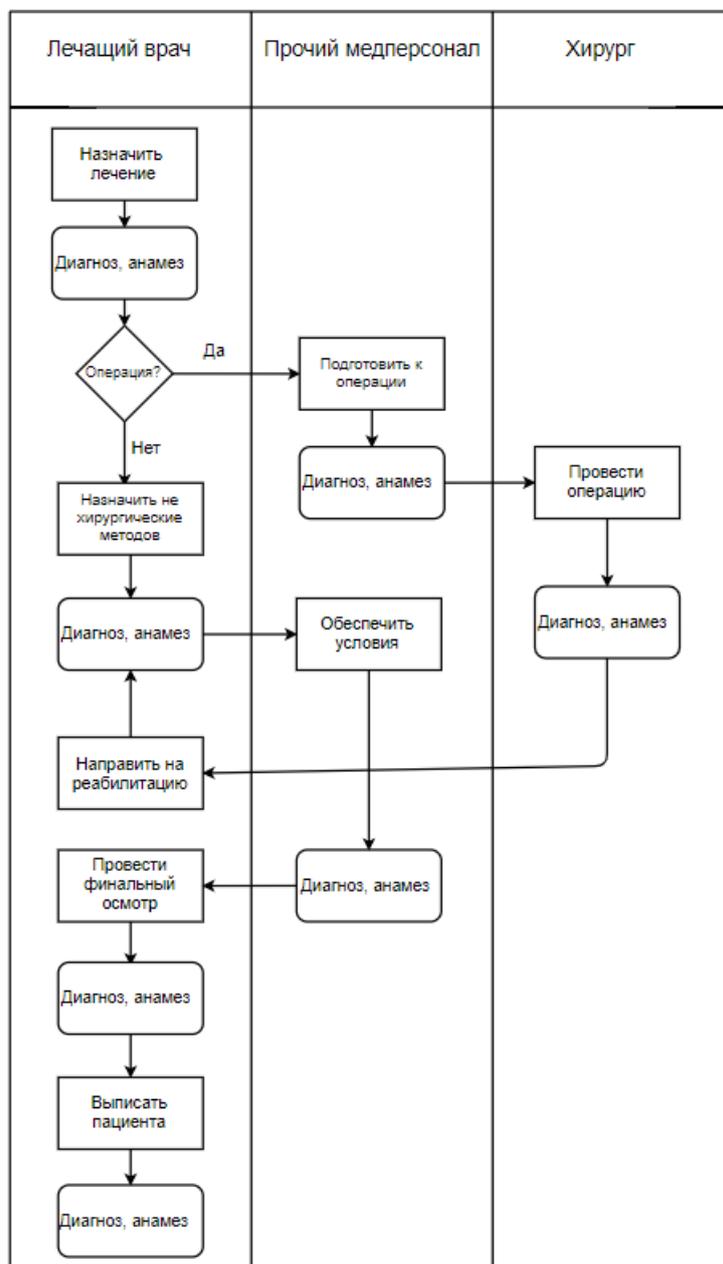


Рис. 4.4. Бизнес-процесс А3.3 - "Провести лечение" на 2-м уровне детализации в модели AS-IS

5. Моделирование процессов с помощью IDEF0 и BPMN SLD в модели TO-BE

Проектирование в модели TO-BE позволяет увидеть, как будут выглядеть ключевые бизнес-процессы медицинского учреждения после внедрения разрабатываемого приложения. Это позволяет разработчику представить схему работы

приложения на конкретном бизнес-объекте. Важной составляющей разрабатываемого приложения является база данных, призванная заменить все бумажные носители информации. Аналогично модели AS-IS опишем ключевые процессы на верхних и нижних уровнях детализации с помощью нотаций IDEF0 и BPMN SLD (рис. 5.1-5.4).

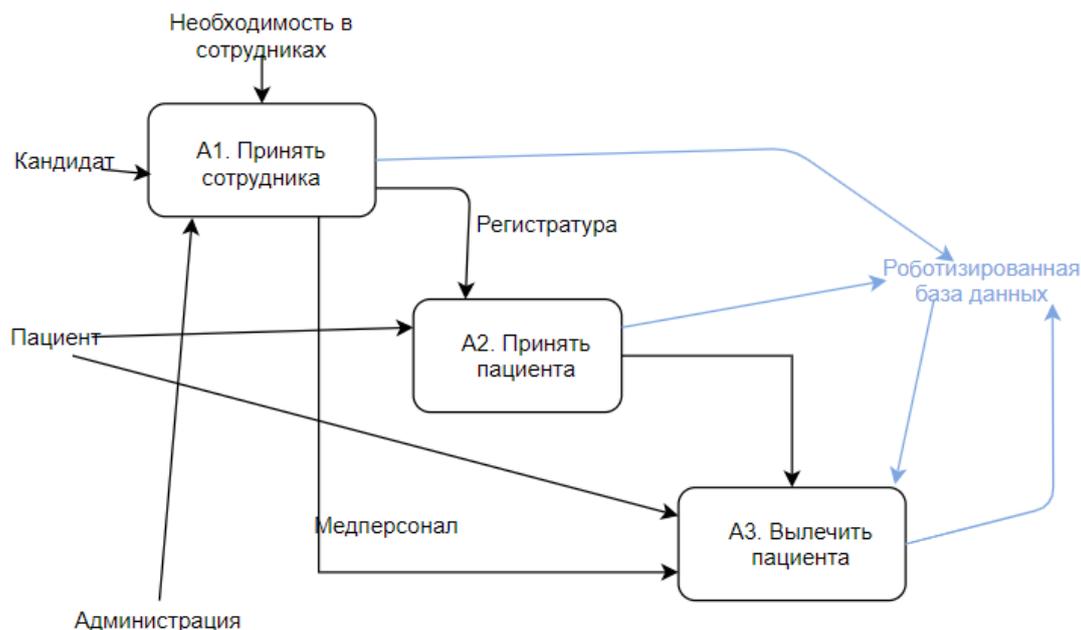


Рис. 5.1. Ключевые бизнес-процессы на начальном уровне описания в модели TO-BE

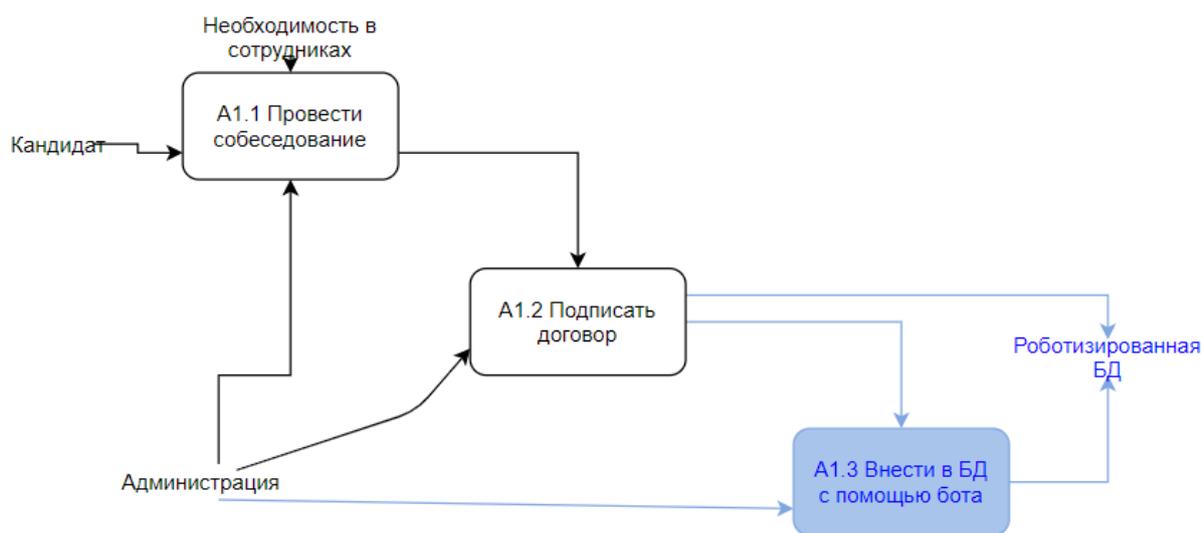


Рис. 5.2. Бизнес-процесс А1 - "Нанять сотрудника" на 1-м уровне детализации в модели TO-BE

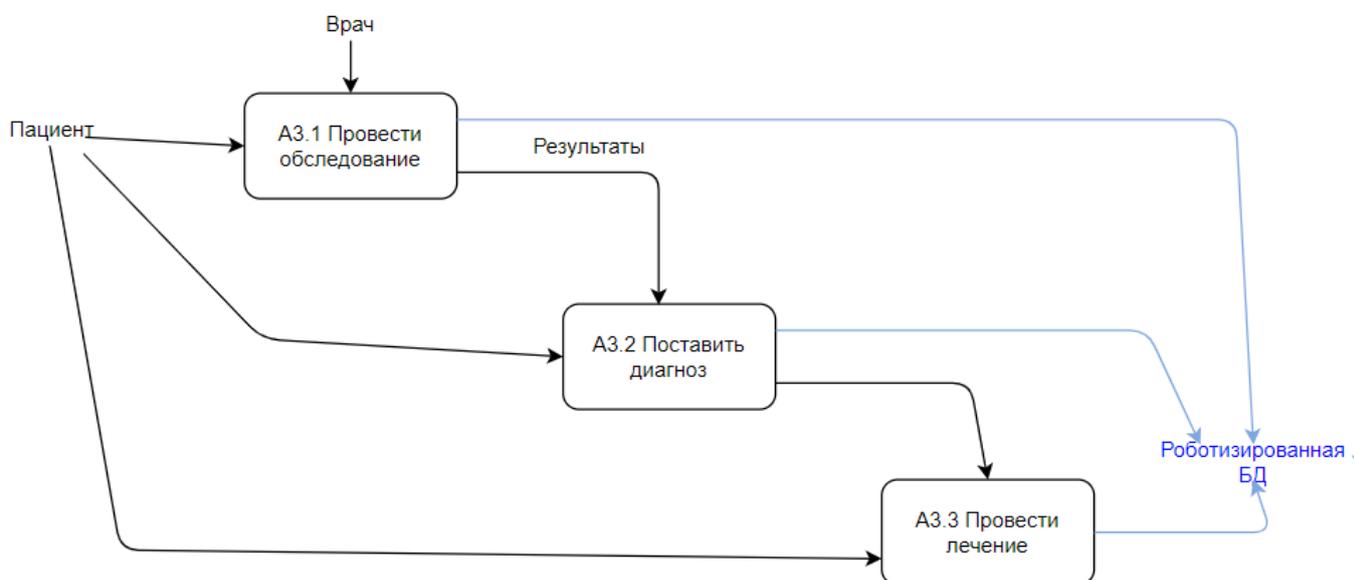


Рис. 5.3. Бизнес-процесс А3 - "Вылечить пациента" на 1-м уровне детализации в модели TO-BE

Как говорилось выше, использование IDEF0 для моделирования бизнес-процессов на более низких уровнях декомпозиции непоказательно, поэтому моделирование проведем в нотации BPMN SLD.

Как видно из рисунков выше, разрабатываемое приложение способно упростить работу медперсонала и избавить его от необходимости заполнять большое число бумажных документов. Карта бизнес-процессов городской больницы является незаменимым инструментом для визуализации функционирования отдельных подпроцессов предприятия (рис. 5.5).

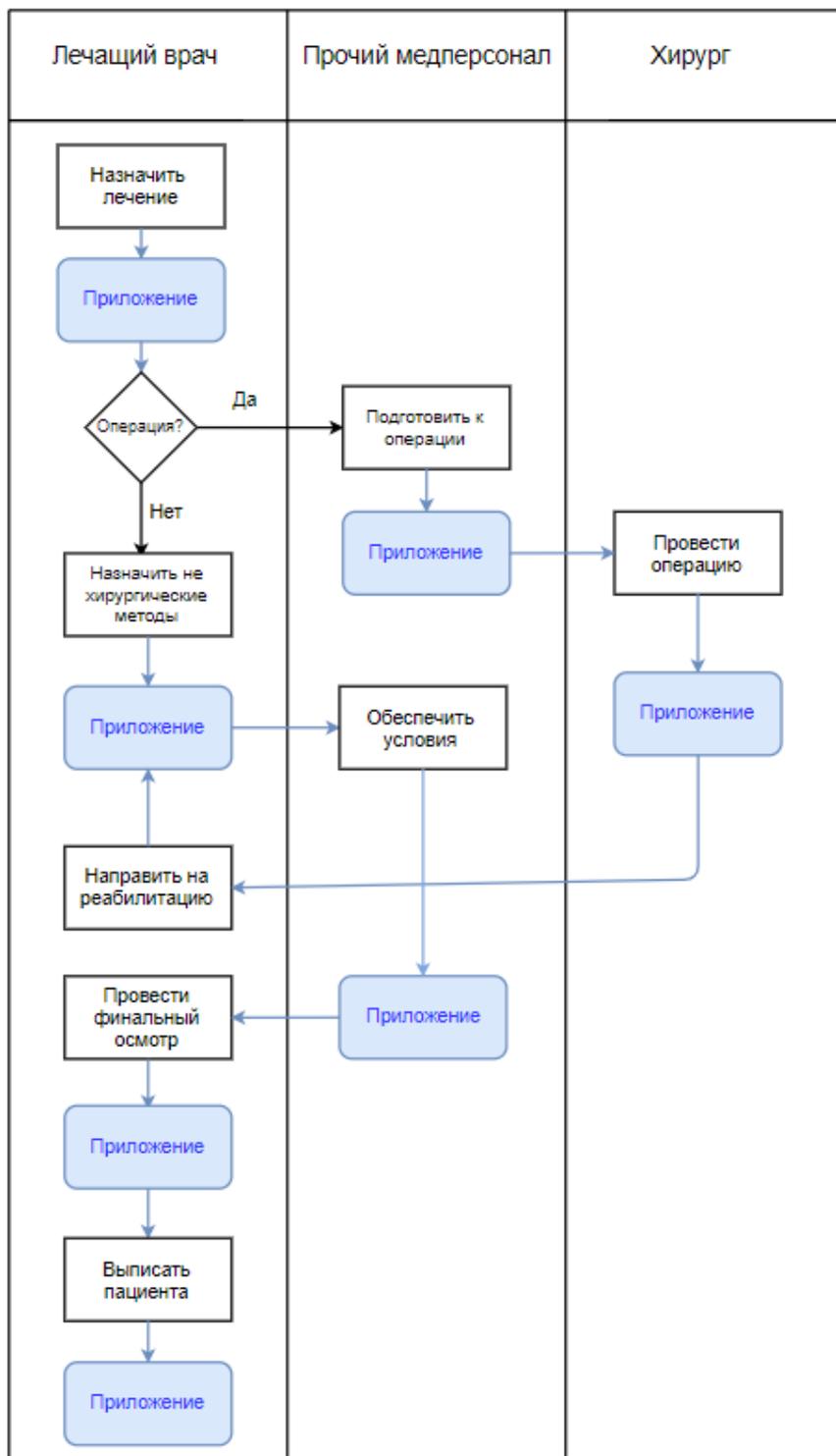


Рис. 5.4. Бизнес-процесс А3.3 - "Провести лечение" на 2-м уровне детализации в модели TO-BE

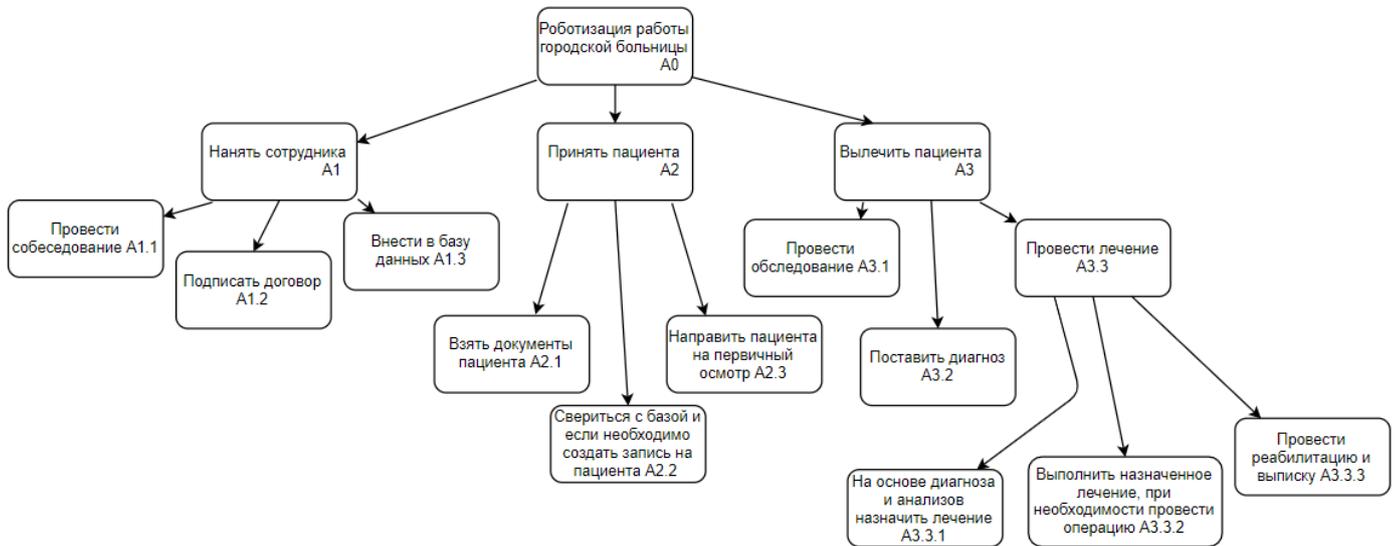


Рис. 5.5. Карта бизнес-процессов в модели TO-BE

6. Разработка программы

На основе проведенного моделирования бизнес-процессов в модели TO-BE была показана целесообразность создания базы данных для оцифровки данных городской больницы с последующей возможностью роботизации части процессов. Разработка будет вестись с использованием спиралевидной модели внедрения, следуя упомянутым в табл.2.1 приоритетам каждого из бизнес-требований.

6.1. Разработка базы данных (1-й виток спирали)

На первом витке разработки разрабатывается структура базы данных, подлежащая нормализации до 3-й нормальной формы [10]. После проектирования структуры и взаимосвязей таблиц, они реализовывались в среде MS Access. Результаты разработки показаны на рисунках 6.1-6.2.

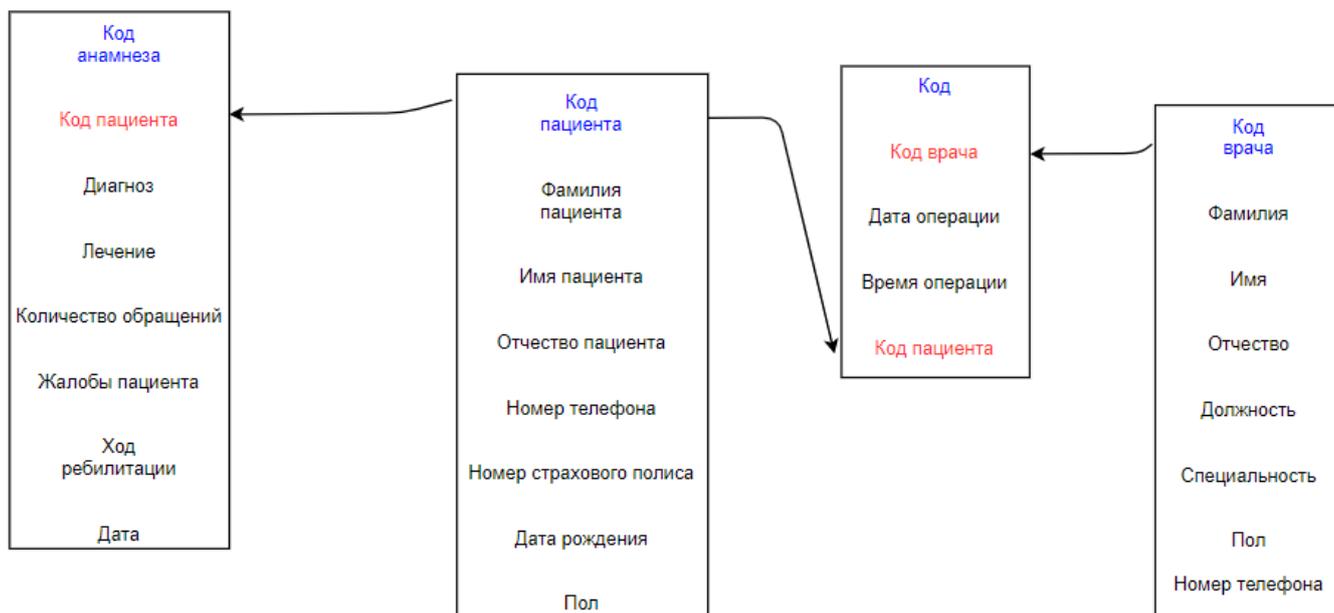


Рис. 6.1. Результаты проектирования структуры таблиц данных

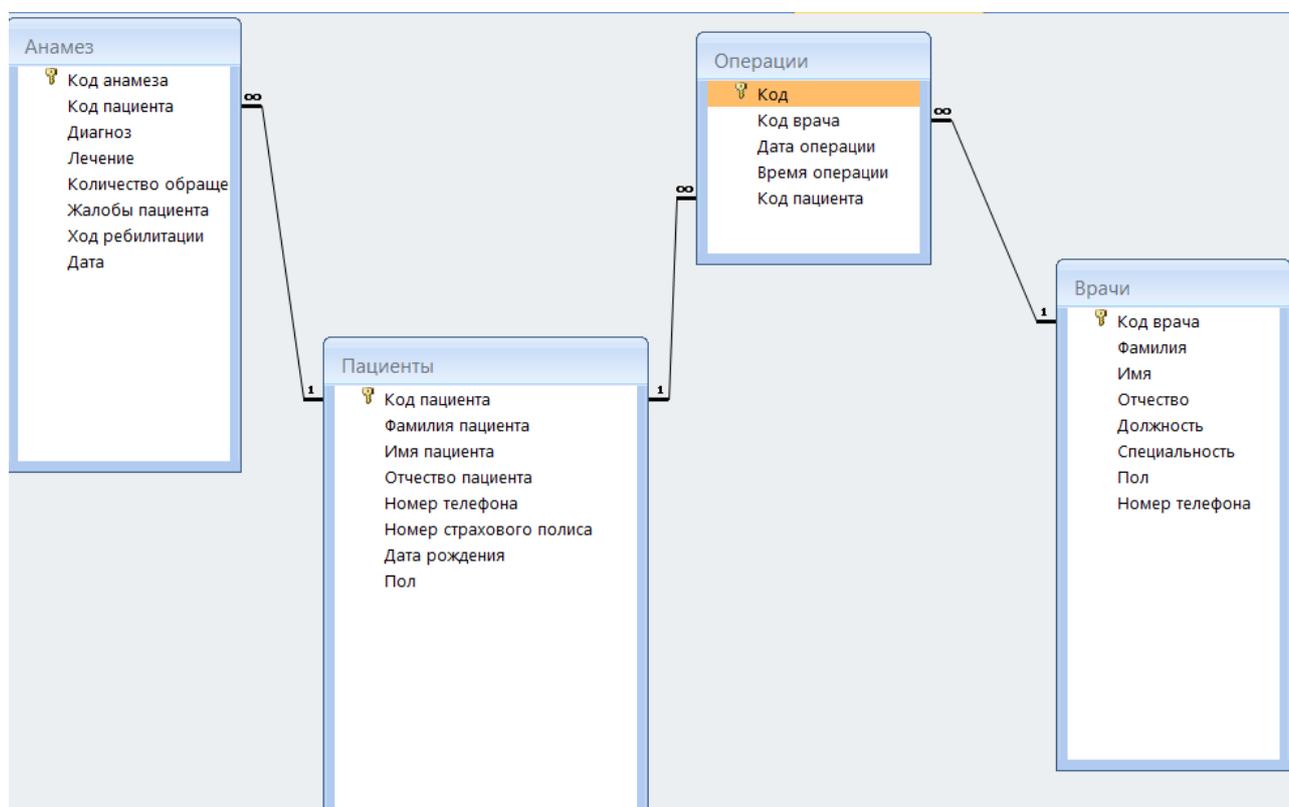


Рис. 6.2. Результаты реализации структуры таблиц в MS Access

6.2. Разработка экрана ввода данных (2-й виток спирали)

Второй цикл разработки позволяет создать пользовательский интерфейс для ввода данных, который обеспечивает взаимодействие пользователя с данными. Этот виток спирали является не менее важным, чем предыдущий: мало просто создать таблицы баз данных, нужно их еще и заполнить, используя пользовательский интерфейс. Итоги реализации графического пользовательского интерфейса даны ниже (рис. 6.3-6.5).

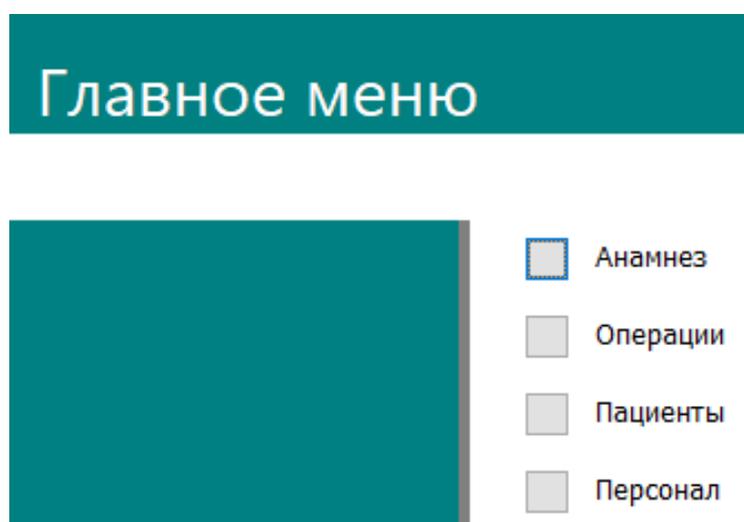


Рис. 6.3. Главное меню программы

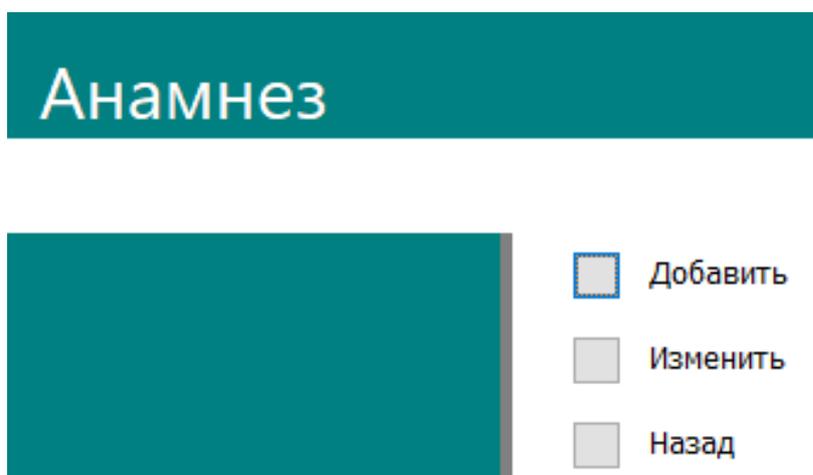


Рис. 6.4. Подменю опции "Анамнез"

Операции

Код (№)	Код врача	Время операции	Код пациента
	Фамилия:	Дата операции	Фамилия пациента:
	Имя:		Имя пациента:
	Отчество:		Отчество пациента:
	Специальность:		



Рис. 6.5. Форма для обработки данных, возникающая при выборе опции "Анамнез" и нажатии пункта "Добавить"

6.3. Разработка элементов роботизации (3-й виток спирали)

На третьем шаге разработки записываются последовательности действий для роботизации отдельных элементов интерфейса по обработке данных. Сначала в UiPath был создан скрипт, открывающий базу данных (рис. 6.6). После этого создавались транзакции для работы с информацией из таблиц баз данных. Была добавлена возможность копировать информацию из электронной таблицы. В завершении строился скрипт по сохранению изменений в базе данных и ее закрытию (рис. 6.7-6.8).

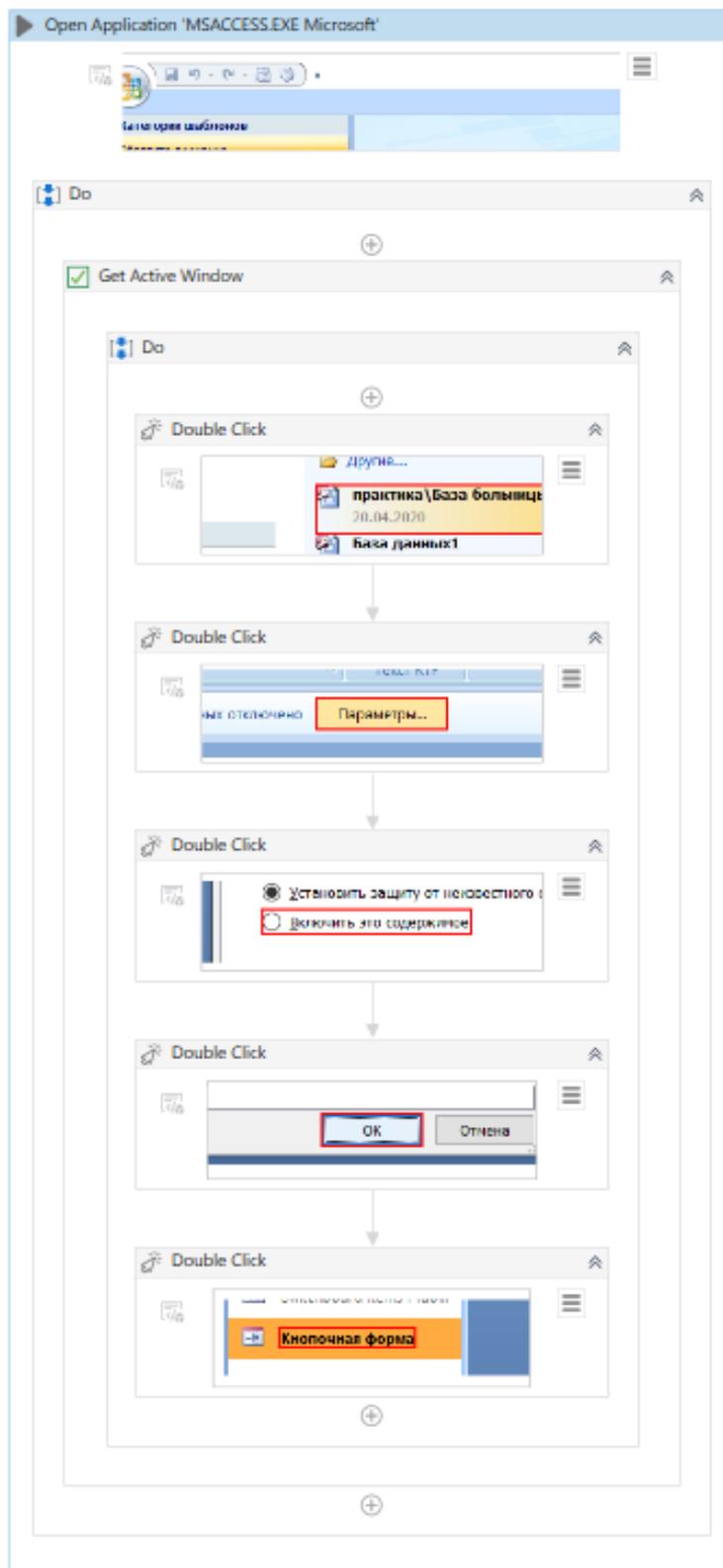


Рис. 6.6. Скрипт в UiPath для открытия базы данных

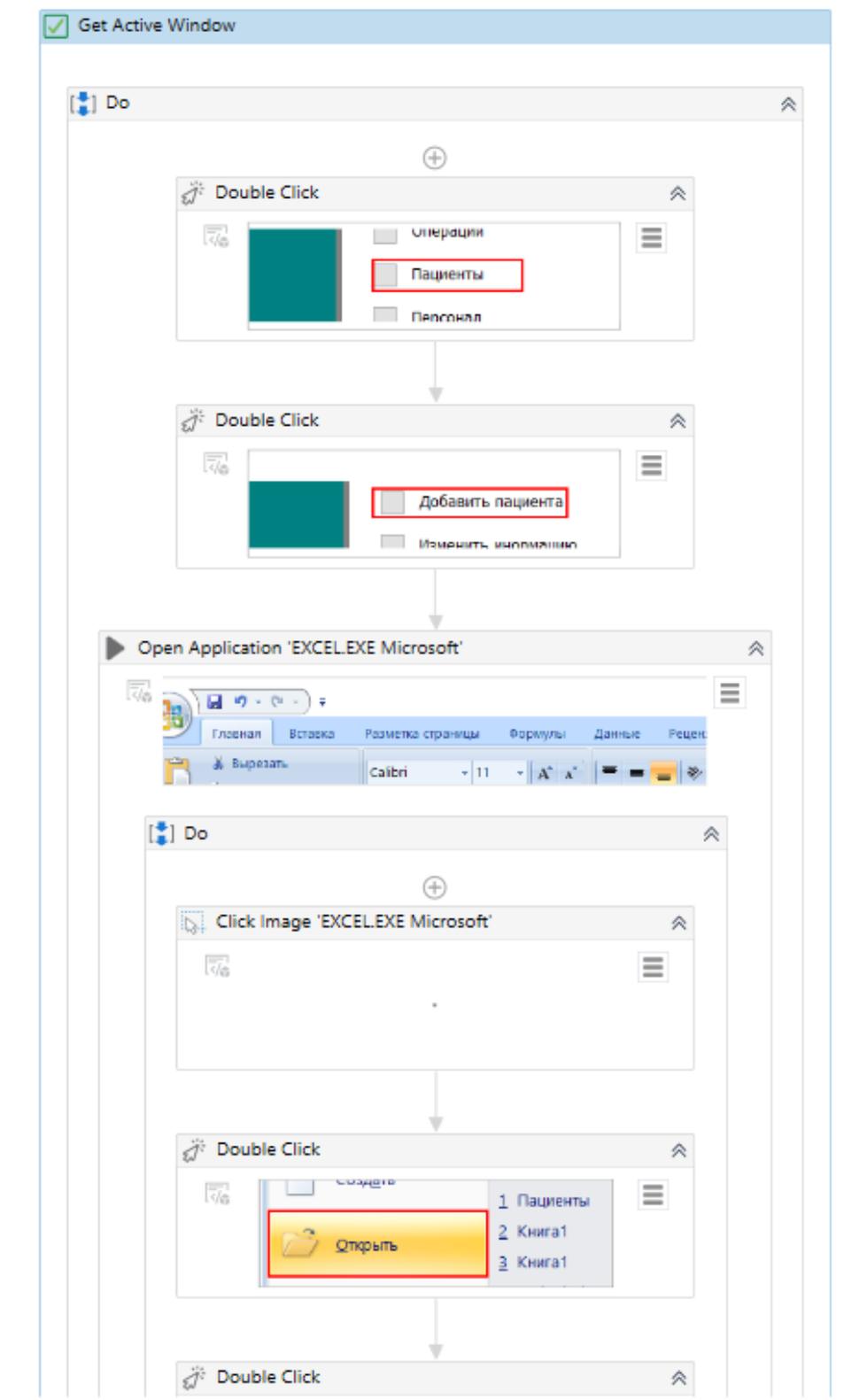


Рис. 6.7. Скрипт для добавления записи в базу данных

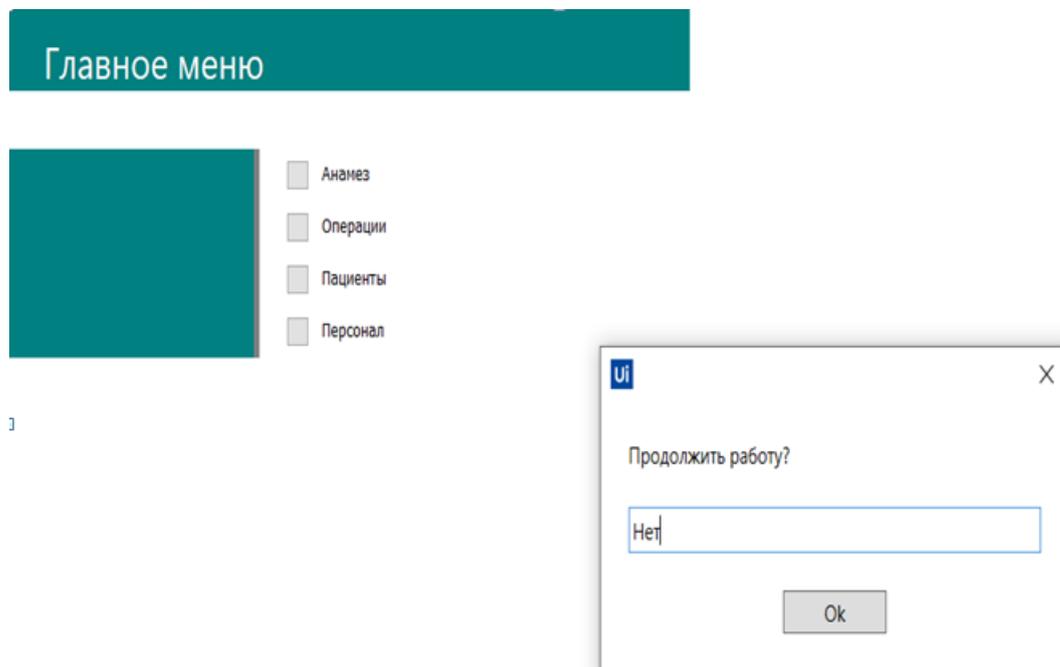


Рис. 6.8. Демонстрация работы скрипта по закрытию базы данных

6.4. Объединение элементов в единое приложение (4-й виток спирали)

Далее созданные элементы управления базой данных из MS Access и скрипты обработки данных объединялись в единую программную разработку при помощи инструмента "Блок-схема" в UiPath. Для работы с каждым пунктом меню и операцией над данными в MS Access была создана отдельная блок-схема в UiPath. Все блок-схемы объединялись в одну единую схему, что дало возможность гибко выбирать между разными опциями обработки данных (рис. 6.9-6.11).

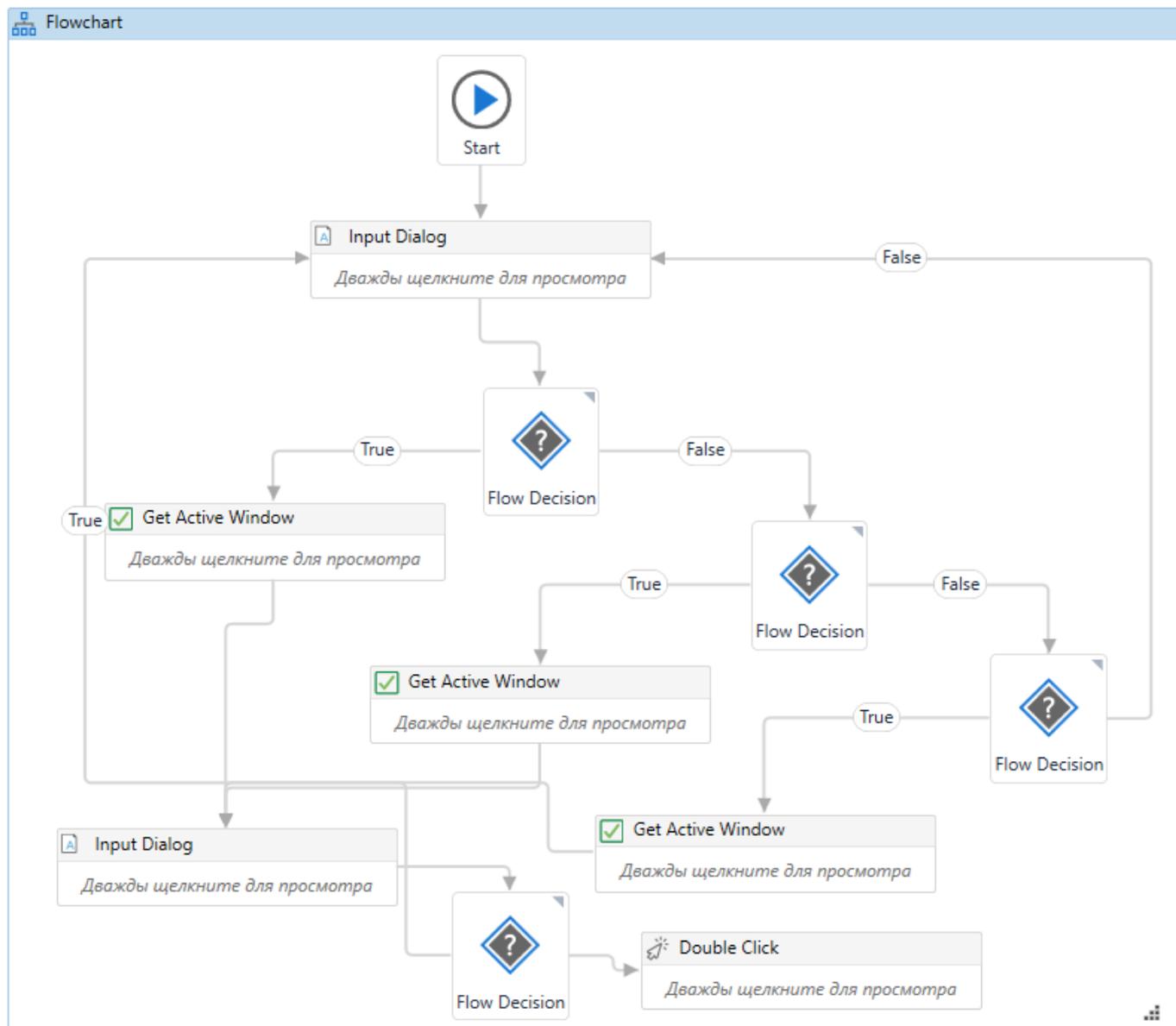


Рис. 6.10. Блок-схем робота, отвечающая за ввод данных в экране "Анамнез"

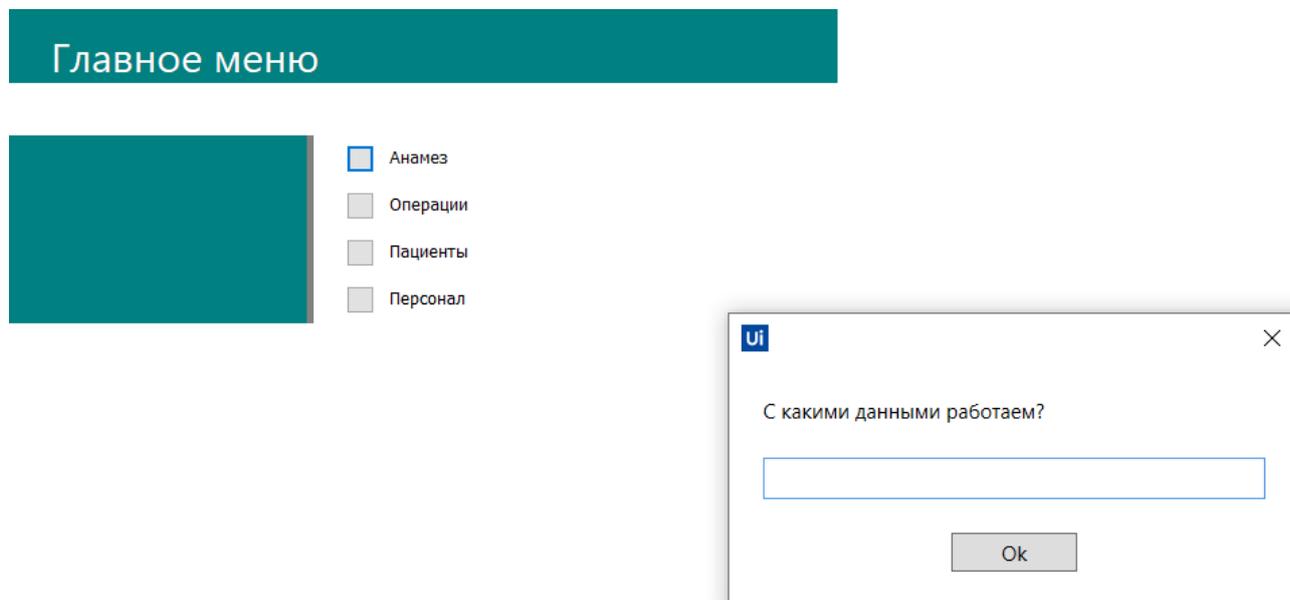


Рис. 6.11. Демонстрация робота, для работы которого необходимо указать путь к Excel-файлу с данными

По результатам реализации 4-х циклов разработки было получено приложение в среде MS Access. Приложение позволяет обрабатывать такие транзакционные сущности как "Анамнез" и "Операции", а также вести основные данные вида "Пациенты" и "Персонал". Над каждой сущностью доступны операции создания, изменения и удаления. В среде UiPath были настроены и записаны скрипты, позволяющие добавлять новые записи в таблицы баз данных MS Access, первоначально хранящиеся в Excel-таблицах. Каждый скрипт UiPath эмулирует запуск экрана ввода данных MS Access и переносит в него информацию из электронной таблицы.

7. Заключение

По итогам работы можно с уверенностью сказать, что переход медицинских учреждений к использованию цифровых решений - это объективная реальность будущего, можно даже сказать, что уже ближайшего настоящего. Ведь это приводит к заметному повышению скорости и эффективности работы медицинского персонала.

Спиралевидная модель внедрения весьма эффективна при разработке приложений, которые задействуют сразу несколько связанных между собой программ. Эта модель позволяет тестировать создаваемый программный продукт в ходе

разработки и сопоставлять результаты с требованиями заказчика, что дает возможность сэкономить значительные средства на окончательной доработке приложения и доведения его до вида, который ожидает клиент.

В процессе реализации проекта приходилось ставить четкие сроки для каждого цикла разработки программного обеспечения. Концепция решения неоднократно была пересмотрена по ходу реализации. Однако, несмотря на эти сложности можно утверждать, что спиралевидная модель весьма удобна при поэтапной разработке объемных приложений.

Практическая часть работы реализовывалась в системе управления базами данных MS Access и на платформе по роботизации процессов UiPath. Данные программы интуитивно понятны и просты для создания стандартных решений по документообороту. В MS Access процесс создания баз данных достаточно прост и логичен. Решение UiPath позволяет генерировать роботов без знания синтаксиса языков программирования. Построение последовательностей и разветвлений действий происходит путем добавления логических блоков, а связи между ними строятся за пару кликов. Данные программные решения сильно упростили создание софта для роботизации городской больницы.

По итогам работы наработан опыт в разработке сложных многоступенчатых программных решений, которые значительно эффективнее не только архаичных методов ведения бизнес-процессов, но и простых решений по автоматизации бизнес-структур. Основной результат работы - понимание того, что система, автоматизирующая работу с данными, позволяет сильно сократить расходы организации. Это возможно путем уменьшения затрат на обработку данных, которые будут вестись не на бумажных носителях, а в разрабатываемом программном продукте.

Литература

1. Бекаревич Ю. Б., Пушкина Н. В. Самоучитель Microsoft Access 2013 - СПб.: БХВ-Петербург - 2014. - 464 с.
2. Обучающий портал по работе с UiPath. UiPath Academy. URL: <https://academy.uipath.com>.
3. Лодон Д., Лодон К. Управление информационными системами. - СПб.: Питер, 2005. - 910 с.

4. Невлюдов И. Ш., Евсеев В. В., Бортникова В. О. Модели жизненного цикла программного обеспечения при разработке корпоративных информационных систем технологической подготовки производства. – 2011.
5. Boehm B.W. A spiral model of software development and enhancement / Boehm B., Egyed A. // IEEE Computer, May 1988, pp. 61-72.
6. Ковалев С., Ковалев В. Секреты успешных предприятий: бизнес-процессы и организационная структура. – М.: БИТЕК, 2012. – 498 с.
7. Степанов Д.Ю. Анализ, проектирование и разработка корпоративных информационных систем: уровень приложений / МГТУ МИРЭА. – М., 2017. URL: <https://stepanovd.com/training/12-erp/52-erp-8-applicationlevel>.
8. Степанов Д.Ю. Анализ, проектирование и разработка корпоративных информационных систем: уровень бизнес-процессов / МГТУ МИРЭА. – М., 2017. URL: <https://stepanovd.com/training/12-erp/51-erp-7-processlevel>.
9. Гвоздева Т.В., Баллод Б.А. Проектирование информационных систем: учебное пособие. – Ростов н/Д.: Феникс, 2009. – 508 с.
10. Дейт К. Дж. Введение в системы баз данных / пер. с англ. и ред. К. А. Птицына – 8-е изд. – М.: Вильямс – 2016. – 327 с.

Выходные данные статьи

Чепров Е.Д. Применение спиралевидной модели внедрения информационных систем для роботизации городской больницы на основе программных решений MS Access и UiPath // Корпоративные информационные системы. – 2021. – №1 (13) – С. 11-36. – URL: <https://corpinfosys.ru/archive/issue-13/112-2021-13-rpa>.

Об авторе



Чепров Евгений Дмитриевич – студент 4-го курса кафедры оптических и биотехнических систем и технологий физико-технологического института РТУ МИРЭА. Тема выпускной квалификационной работы бакалавра «Роботизация работы городской больницы на основе спиралевидной модели внедрения информационных систем». Электронная почта: mail@corpinfosys.ru