Оглавление

8.5. Организация сбора данных 2
8.5.1. Создание опросов с SurveyDown 2
8.5.2. Система электронной ИРК LibreClinica 18

8.5. Организация сбора данных

Ключевой задачей при проведении исследований является сбор данных. При этом саму процедуру сбора данных с точки зрения сложности исполнения можно условно разделить на два типа: опрос и исследование. Опрос можно охарактеризовать, как однократное заполнение формы с вопросами, возвращение к которой не планируется и проводится анонимно. Обычно электронная ссылка на опрос широко распространяется в Интернете, и участие в нем доступно для всех желающих. Результаты опроса доступны исключительно его создателю, который также имеет право закрыть опрос для дальнейшего приема ответов.

Исследование - это скорее сбор данных по пациенту/эпизоду заболевания/выделенному возбудителю и т.д., который подразумевает протяженный во времени процесс. В этом случае предполагается возвращение к уже внесенной ранее информации и ее изменение, дополнение, удаление. Информация в этом случае обычно предназначена для ограниченного круга лиц и для работы с ней требуется авторизация.

На первый взгляд существуют уже готовые инструменты для организации такого сбора информации: для опросов можно использовать Яндекс.Формы (https://forms.yandex.ru), для исследований EpiCollect5 (https://five.epicollect.net). Однако оба этих сервиса работают только в сети Интернет и хранят собранные данные на собственных серверах. В ситуациях, требующих полного контроля над процессом хранения данных и размещения их на собственных серверах, а также возможности сбора данных в изолированной сетевой среде без доступа к внешнему Интернету, упомянутые ранее инструменты становятся неприменимы. Для решения поставленных задач далее будут рассмотрены два инструмента, которые удовлетворяют вышеописанным ограничениям.

8.5.1. Создание опросов с SurveyDown

SurveyDown (https://surveydown.org) — это платформа для создания опросов с помощью языка R, Quarto и Shiny. Она может быть как размещена в сети Интернет, так и запущена локально. Данные хранятся в базе данных PostgreSQL (**см. раздел 3.2.**).

Для создания опроса с помощью данной технологии используется R и Markdown, а все содержимое хранится в двух файлах:

- *survey.qmd* - документ Quarto (**см. раздел 8.3.**), содержащий непосредственно опрос (страницы, вопросы и т. д.);
- *app.R* - скрипт R, определяющий Shiny-приложение (**см. раздел 8.4.**), которое содержит глобальные настройки (библиотеки, конфигурацию базы данных и т. д.) и параметры конфигурации сервера.

Благодаря такому подходу опрос можно легко воспроизвести, поделиться им и управлять его версиями с помощью Git (**см. раздел 8.1.**), а также одновременно работать нескольким пользователям над созданием опроса.

Для установки SurveyDown необходимо в R выполнить следующую команду:

```
install.packages("surveydown")
```

Далее необходимо создать новый проект опроса с помощью команды *sd_create_survey()*. В качестве аргумента следует указать папку, где будет создан проект. По умолчанию он будет создан в текущей папке.

```
surveydown::sd_create_survey(".")
```

Шаблон созданного проекта содержит несколько страниц с вопросами и находится в состоянии готовности к развертыванию. Для запуска проекта в RStudio следует открыть файл *app.R* и нажать кнопку "Запустить приложение". Пример содержимого файла *app.R* представлен далее.

```
library(surveydown)

# Настройки подключения к базе данных.
db <- sd_database(
  host   = "",
  dbname = "",
  port   = "",
  user   = "",
  table  = "",
  # В режиме разработки вместо базы данных будет использоваться CSV-файл.
  ignore = TRUE
)

# Определение серверной части приложения.
server <- function(input, output, session) {
  # Описание логики для пропуска вопросов.
  sd_skip_if()
  # Описание условий для показа вопросов.
  sd_show_if()
  # Настройки приложения.
  sd_server(
    db = db
  )
}
# Запуск приложения shiny.
shiny::shinyApp(ui = sd_ui(), server = server)
```

Сами вопросы определяются в файле *survey.qmd*. Вопросы разбиты на страницы, каждая из которых определяется блоками :::, и может содержать идентификатор, который указывает, что страница является опросной (*.sd-page*). Внутри страниц определяются вопросы в виде чанков кода на R и функции *sd_question()*. В зависимости от параметров, переданных в функцию определяется тип вопроса. С полным списоком типов вопросов можно ознакомиться по ссылке https://surveydown.org/question-types.

Каждая страница должна завершаться кнопкой - перехода к следующей/предыдущей странице или завершения опроса. Внутри самой страницы можно размещать содержимое на языке Markdown (**см. раздел 8.2.7**).

Пример кода первой страницы приведен далее.

```
::: {#welcome .sd-page}

# Welcome to our survey!

This is a simple demonstration of a surveydown survey. It has two pages with one
question on each page.

Here is a basic "multiple choice" question, created using `type = 'mc'` inside the
`sd_question()` function:

```{r}
sd_question(
  type  = 'mc',
  id    = 'penguins',
  label = "Which type of penguin do you like the best?",
  option = c(
    'Adélie'    = 'adelie',
    'Chinstrap' = 'chinstrap',
    'Gentoo'    = 'gentoo'
  )
)
```

You need to insert next buttons with `sd_next()` and set the `next_page` argument
to the name of the page you want to go to next.

```{r}
sd_next(next_page = 'page2')
```

:::
```

При запуске данной страницы будет получено следующее содержимое.

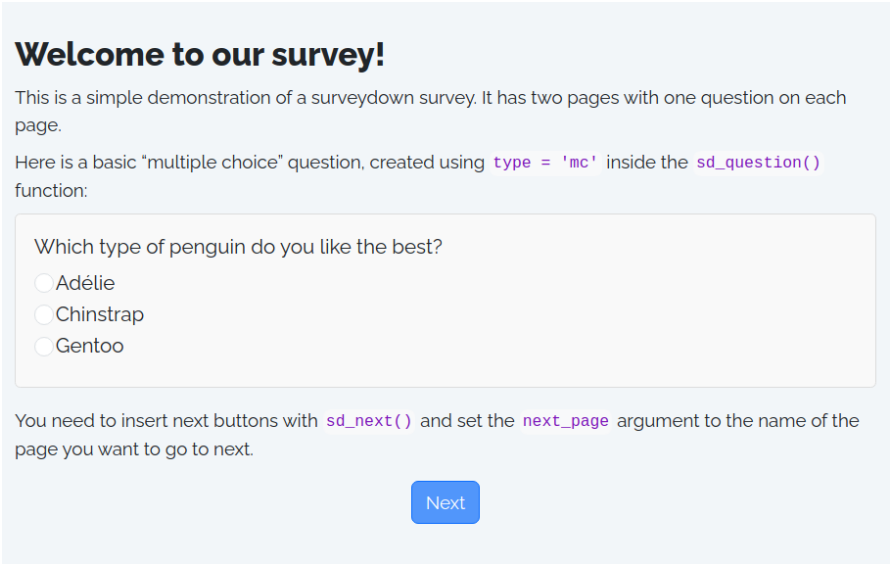


Рисунок 8.201. Страница опросника из стартового шаблонного проекта

Следует отметить, что по умолчанию в приложении используется английский язык. И если содержимое самой страницы можно переписать, то названия функциональных кнопок по умолчанию останутся неизменными. Чтобы локализовать приложение на нужном языке, нужно создать дополнительный файл *translations.yml* и определить там подписи функциональных элементов опросника для русского языка.

```
surveydown::sd_create_translations(language = 'ru')
```

Содержимое файла translations.yml может быть примерно следующим.

```
ru:
  cancel: Отмена
  confirm_exit: Подтвердите Выход
  sure_exit: Вы хотите завершить опрос?
  submit_exit: Отправить и закрыть
  warning: Внимание!
  required: Пожалуйста, заполните все обязательные вопросы, чтобы продолжить
  rating_title: Перед тем как уйти
  rating_text: Оцените ваше впечатление от опроса
  rating_scale: от 1-плохо до 5-превосходно
  next: Следующая страница
  exit: Завершить опрос
  close_tab: Пожалуйста, закройте вкладку, чтобы завершить опрос.
  choose_option: Выберите вариант из предложенных
  click: Нажмите сюда
  redirect: Перейти к
  seconds: секунд
  new_tab: Открыть в новой вкладке
  redirect_error: 'Ошибка: Этот текст не ведет ни к какому вопросу'
```

В данном файле можно определять несколько языков и устанавливать нужный при запуске приложения. В данном примере будет использован русский язык (ru), поэтому нужно сообщить об этом приложению в файле *app.R*.

```
server <- function(input, output, session) {
  # ...
  sd_server(
    db = db,
    language = "ru"
  )
}
```

Последующая модификация опросника включает преобразование стартовой страницы "welcome" в приветственное сообщение, которое будет непосредственно перенаправлять респондентов к вопросам. При этом следует отметить, что для кнопки перехода можно задавать индивидуальные подписи и указывать, на какую страницу она ведет.

```
::: {#welcome .sd-page}

# Добро пожаловать в опросник нашего исследования!

```{r}
sd_next(label = "Начать опрос", next_page = "patient")
```

:::
```

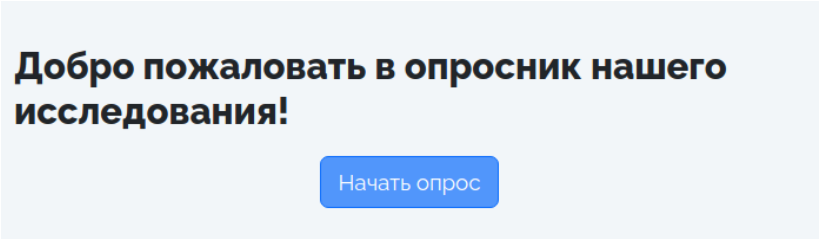


Рисунок 8.202. Результат модификации стартовой страницы опросника

Далее будет создана страница *patient* с несколькими вопросами:

- Инициалы - текстовое поле;
- Дата рождения - выбор даты;
- Пол - выбор одного варианта из списка;

Каждый вопрос будет определяться одной функцией *sd_question()*, но различным набором параметров. Важно, чтобы параметр *id* в функции *sd_question()* был уникальным - именно по этому идентификатору вопроса к нему можно будет обращаться в коде и с таким именем будут сохраняться результаты.

```
::: {#patient .sd-page}

## Данные о пациенте

Заполните, пожалуйста данные о себе:

```{r}
```
```

```
sd_question(
  type  = 'text',
  id    = 'initials',
  label = "Ваши инициалы (ФИО)"
)

sd_question(
  type  = 'date',
  id    = 'dob',
  label = "Дата рождения"
)

sd_question(
  type   = 'mc',
  id     = 'sex',
  label  = "Пол",
  option = c(
    "Мужской" = "male",
    "Женский" = "female"
  )
)

sd_next(next_page = 'diag')
```

:::

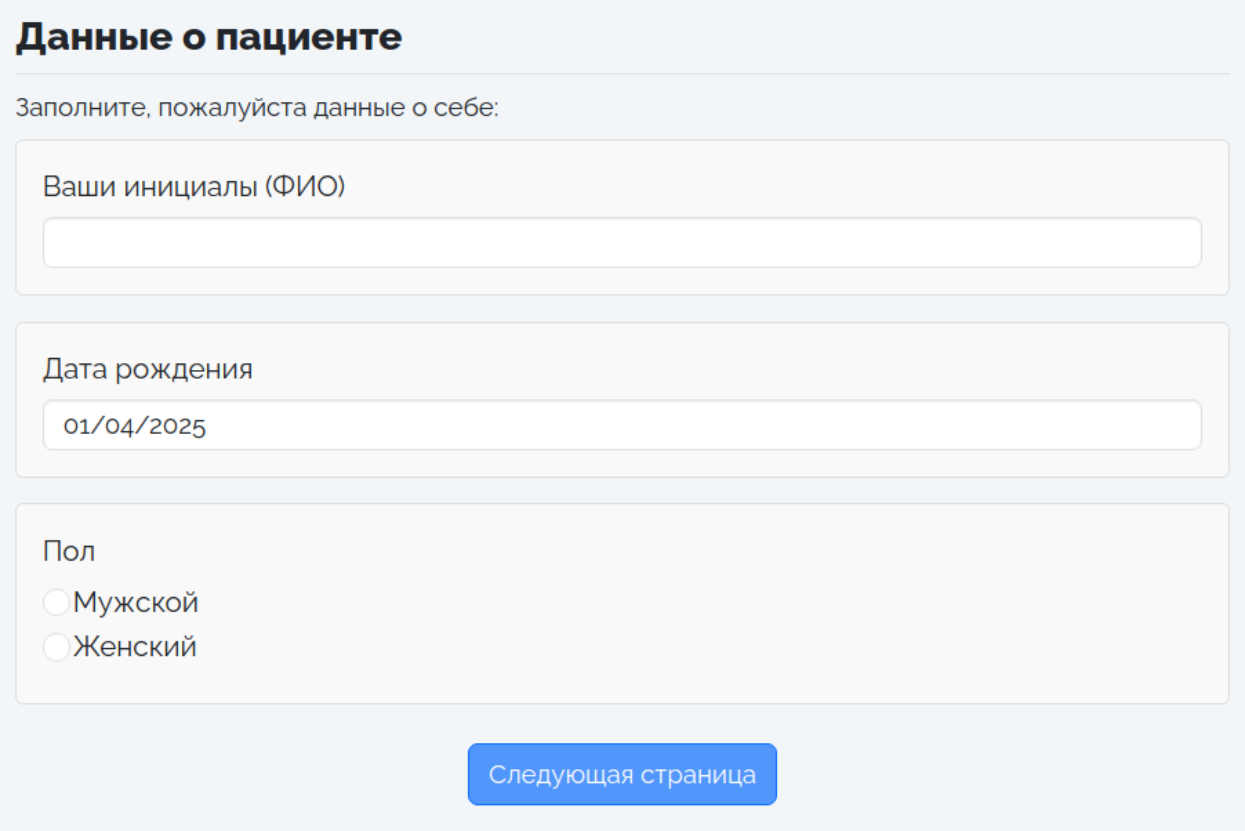


Рисунок 8.203. Созданная страница с данными пациента

Далее будет добавлена логика показа отдельных вопросов. Например, при выборе женского пола будет показываться дополнительный вопрос “Наличие беременности”, и в случае положительного ответа будет предлагаться указать срок беременности в неделях. Для этого в *survey.qml* необходимо создать еще два вопроса.

Для вопроса с выбором из нескольких вариантов необходимо задать пары значений в параметр *option*: значение слева от знака “=” будет отображаться на странице, значение справа - записываться в базу данных.

```
sd_question(
  type   = 'mc',
  id     = 'pregnancy',
  label  = "Наличие беременности?",
  option = c(
    "Да" = "yes",
    "Нет" = "no"
  )
)

sd_question(
  type  = 'numeric',
  id    = 'pregnancy_weeks',
  label = "Укажите количество недель"
)
```

Одновременно с добавлением вопросов необходимо в файле *app.R* с помощью функции *sd_show_if()* определить условия показа данных вопросов. Дополнительно следует указать, что вопросы “Дата рождения” и “Пол” являются обязательными для заполнения в блоке *sd_server()*.

```
sd_show_if(
  input$sex == "female" ~ "pregnancy",
  input$pregnancy == "yes" ~ "pregnancy_weeks"
)

sd_server(
  db = db,
  language = "ru",
  required_questions = c("sex", "dob")
)
```

В результате добавления данного кода у вопросов “Дата рождения” и “Пол” появятся красные звездочки, а при выборе нужных ответов появятся дополнительные вопросы “Наличие беременности?” и “Укажите количество недель”.

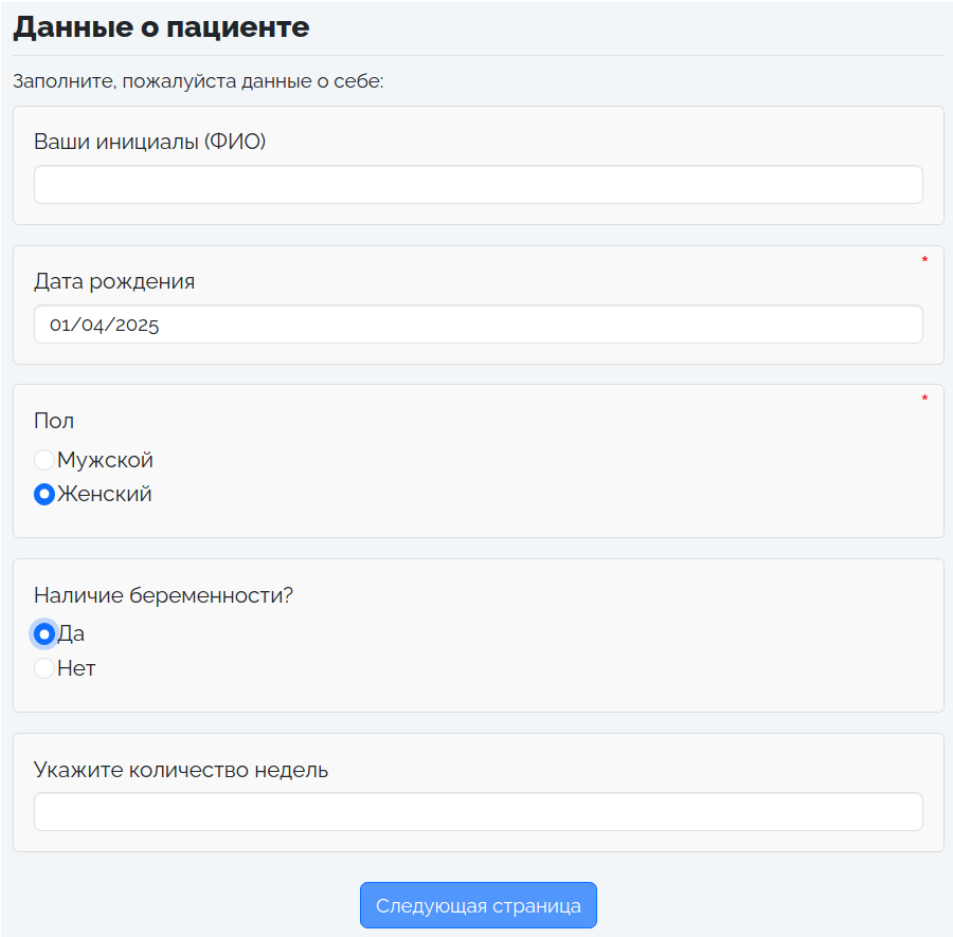


Рисунок 8.204. Созданная страница с данными пациента и дополнительными вопросами

Далее будет добавлена следующую страницу с вопросом о диагнозе пациента. Данный вопрос предполагает возможность выбора нескольких из предложенных вариантов.

```
::: {#diag .sd-page}

## Диагноз

```{r}
sd_question(
  type  = 'mc_multiple',
  id    = 'diag',
  label = "Укажите основные диагнозы?",
  option = c(
    "Интерстициальный цистит (хронический)" = "N30.1",
    "Инфекция мочевыводящих путей без установленной локализации" = "N39.0",
    "Необструктивный хронический пиелонефрит, связанный с рефлюксом" = "N11.0",
    "Острый тубулоинтерстициальный нефрит" = "N10",
    "Острый цистит" = "N30.0",
    "Другой" = "OTHER"
  )
)

sd_next(next_page = 'end')
```
```

```
:::
```

По умолчанию кнопка в конце данной страницы будет вести на страницу завершения опроса. Для примера будет добавлена следующая логика: если респондент выбирает вариант “Острый цистит”, то будет осуществлен переход на дополнительную страницу со шкалой оценки симптомов острого цистита (ACSS) (https://www.zdorovieinfo.ru/wp-content/uploads/2020/12/cistit.pdf).

Для реализации такого перехода необходимо в файле *app.R* в функции *sd_skip_if()* описать условие: переходить на страницу *acss*, если есть хотя бы один вариант ответа “N30.0”.

```
sd_skip_if(
  any(input$diag %in% c("N30.0")) ~ "acss"
)
```

Далее необходимо составить непосредственно страницу с данными опросником. Для примера будет использовано лишь несколько вопросов из данного опросника. В конце страницы для наглядности будет выведено количество баллов, полученных согласно опроснику ACSS.

```
::: {#acss .sd-page}

## Шкала оценки симптомов острого цистита (ACSS)

Пожалуйста укажите, отмечали ли Вы следующие симптомы в течение последних 24 часов, и оцените степень
их выраженности

::: {#diag .sd-page}

## Диагноз

```{r}
sd_question(
  type  = 'mc_multiple',
  id    = 'diag',
  label = "Укажите основные диагнозы?",
  option = c(
    "Интерстициальный цистит (хронический)" = "N30.1",
    "Инфекция мочевыводящих путей без установленной локализации" = "N39.0",
    "Необструктивный хронический пиелонефрит, связанный с рефлюксом" = "N11.0",
    "Острый тубулоинтерстициальный нефрит" = "N10",
    "Острый цистит" = "N30.0",
    "Другой" = "OTHER"
  )
)

sd_next(next_page = 'end')
```

:::
```

Чтобы выводить какие-либо вычисленные на стороне приложения значения необходимо использовать функцию *sd_output()*. С ее помощью можно выводить как значения, так и динамически формируемые вопросы. Для этого в файле *app.R* необходимо добавить отслеживание изменений с помощью функции *observe()*. Далее представлен пример вычисления суммарного количества баллов для отображения на странице.

```
server <- function(input, output, session) {
  # ...
  observe({
    # Вычисление суммарного индекса на основе ввода пользователя.
    acss_1_1 <- as.numeric(input$acss_1_1)
    acss_1_2 <- as.numeric(input$acss_1_2)
    acss_2 <- as.numeric(input$acss_2)
    acss_3 <- as.numeric(input$acss_3)
    acss_4 <- as.numeric(input$acss_4)
    acss_5 <- as.numeric(input$acss_5)
    acss_6 <- as.numeric(input$acss_6)
    acss_sum <- acss_1_1 + acss_1_2 + acss_2 +
      acss_3 + acss_4 + acss_5 + acss_6
    # Сохранение значения в переменной, доступной на странице.
    sd_store_value(acss_sum)
  })
  # ...
}
```

Тем не менее, вычисленное значение отображается только на странице и не может быть сохранено в базе данных, поскольку сохраняются исключительно ответы на вопросы. Для преодоления этого ограничения можно создать динамический вопрос и присвоить ему вычисленное значение. Наиболее подходящим типом вопроса для данного сценария является слайдер, поскольку в отличие от других типов вопросов для него можно задать значение по умолчанию. После вычисления значения формируется список возможных вариантов, состоящий из одного элемента, и устанавливается текущее значение вопроса.

```
# Формирование предзаполненненого вопроса, чтобы сохранить в базу данных.
if (length(as.character(acss_sum)) > 0) {
  options <- c(as.character(acss_sum))
  names(options)[1] <- as.character(acss_sum)
  sd_question(
    type   = "slider",
    id     = "acss_summary",
    label  = "Суммарное количество баллов",
    option = options,
    selected = as.character(acss_sum)
  )
}
```

Непосредственно на странице такой вопрос можно отобразить с помощью функции *sd_output()*, но уже с другими параметрами.

```{r}
sd_output(id = "acss_summary", type = "question")
```

```

В результате такой модификации кода получится следующая страница с опросником (**рисунок 8.205**).

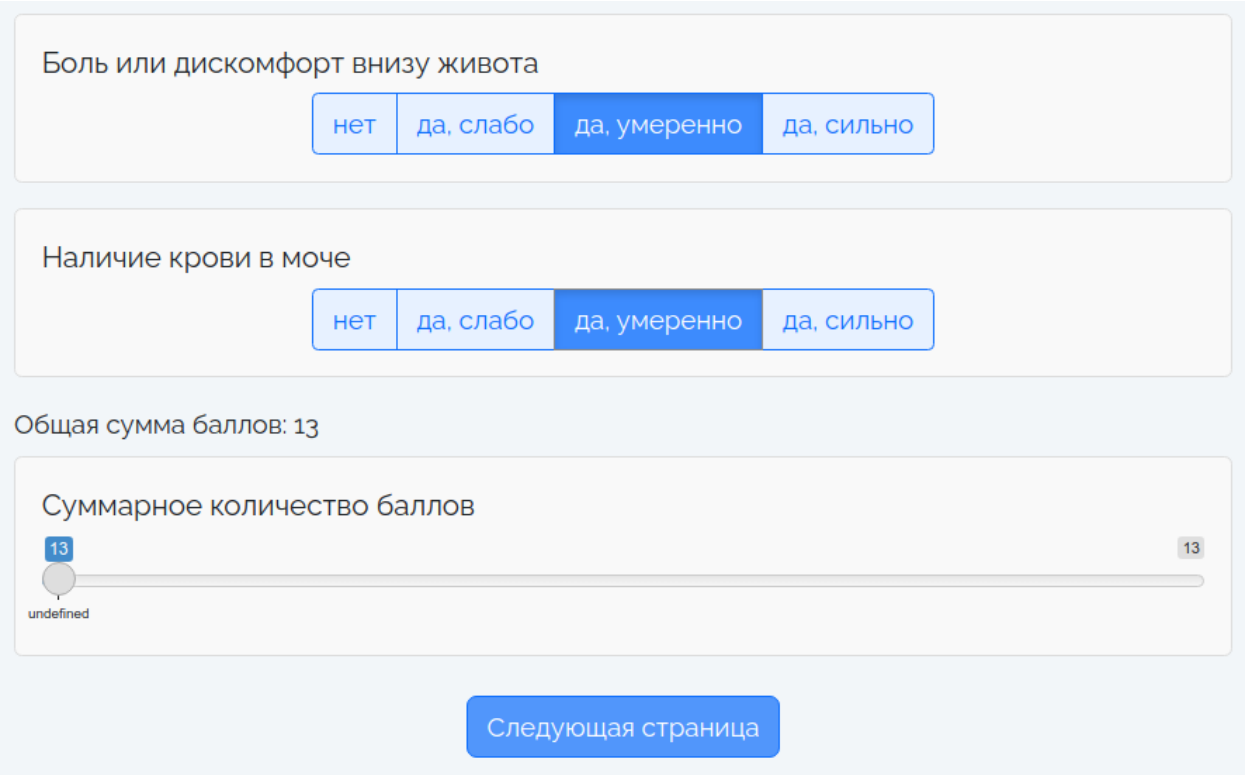



**Рисунок 8.205**. Созданная страница с вопросом-слайдером

Последняя страница в разрабатываемом опроснике будет содержать текст благодарности и кнопку закрытия страницы. Именно по нажатии этой кнопки данные заполненного опросника сохраняются в базу данных.

```
::: {#end .sd-page}

Спасибо за прохождение опроса

Ваши ответы очень важны для нас.

```{r}
sd_close("Закрыть страницу")
```

:::
```

Для обеспечения возможности выгрузки заполненных ответов из браузера необходимо добавить в опросник режим администратора и установить пароль. Для этого нужно выполнить следующую команду:

```
sd_set_password("password")
```
```

Кроме того, необходимо добавить новый параметр в определение сервера в файле *app.R*.

```
sd_server(
  db = db,
  language = "ru",
  required_questions = c("sex", "dob"),
  admin_page = TRUE
)
```

В качестве пароля должен быть указан пароль, который используется для подключения к базе данных. После этого в запущенном приложении по адресу формата *адрес-приложения-в-интернете?admin* становится доступна страница, через которую можно просматривать заполненные ответы и скачивать их в виде файла.

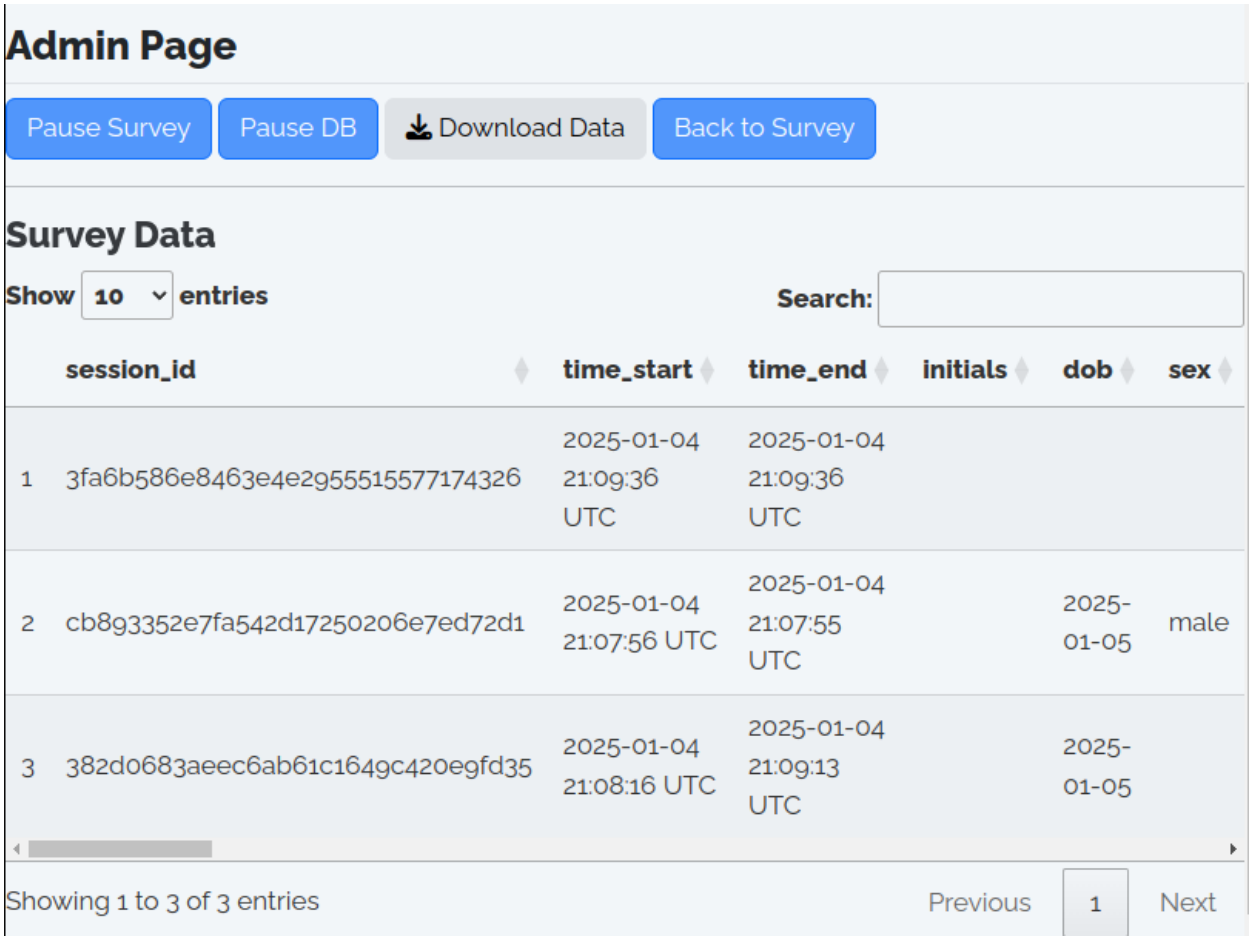


Рисунок 8.206. Страница с ответами

Публикация приложения SurveyDown

По умолчанию рекомендуемый способ развертывания созданного приложения заключается в публикации приложения на сервис Shinyapps.io с использованием бесплатной базы данных сервиса Supabase https://supabase.com. Процедура подробно описана в официальной документации, доступной по ссылке

https://surveydown.org/deployment. Следует отметить, что перед публикацией необходимо создать экземпляр базы данных в сервисе Supabase согласно инструкции https://surveydown.org/store-data и заполнить необходимые параметры подключения к базе данных в файле *app.R*.

```
db <- sd_database(
  host   = "database_address",
  dbname = "database_name",
  port   = "database_port",
  user   = "database_user",
  table  = "tablename",
  ignore = FALSE
)
```

При этом допускается использование одной базы данных для нескольких приложений с опросами, принципиальным моментом является различное наименование создаваемых таблиц в базе данных.

В представленном примере будет рассмотрена публикация приложения с опросником с использованием собственного сервера базы данных PostgreSQL (**см. раздел 3.2.**) и технологии Docker Compose https://docs.docker.com/compose. По умолчанию она уже доступна в установленном Docker (**см. раздел 8.4.5.**).

Технология Docker Compose предоставляет минимальные возможности оркестрации контейнеров (т.е. управлением множеством контейнеров). Она позволяет в одном файле описать параметры запуска и взаимодействия нескольких Docker-контейнеров с возможностью их последовательного или одновременного запуска.

Перед тем как описать создаваемое приложение в Docker Compose, прежде всего необходимо собрать образ с приложением.

Зафиксировать используемое окружение, все зависимости и пакеты в проекте необходимо с помощью команды *renv::init()*. После инициализации нужно выполнить команду *renv::snapshot()*. Все результаты будут записаны в файл *renv.lock*. Далее файл *renv.lock*, содержащий информацию об используемых пакетах и их версиях, будет скопирован “внутрь” создаваемого образа Docker и использован для загрузки и восстановления всех пакетов необходимых версий. Ниже представлен пример Dockerfile для разрабатываемого Shiny-приложения.

```
FROM rocker/shiny:4.4.1
RUN apt-get update -qq && apt-get -y --no-install-recommends install \
    curl \
    libxml2-dev \
    libpq-dev libssl-dev \
    && rm -rf /var/lib/apt/lists/* \
    && curl -L $(curl https://quarto.org/docs/download/_prerelease.json | grep -oP
"(?<=\"download_url\":\s\")https.*${ARCH}\.deb") -o /tmp/quarto.deb \
    && dpkg -i /tmp/quarto.deb \
    && rm /tmp/quarto.deb
```

```
RUN apt-get update && \
    apt-get upgrade -y && \
    apt-get clean
COPY ./renv.lock ./renv.lock
COPY . ./app
RUN Rscript -e 'install.packages("renv")'
RUN Rscript -e 'renv::restore()'
EXPOSE 3838
CMD ["R", "-e", "shiny::runApp('/app', host = '0.0.0.0', port = 3838)"]
```

Содержимое данного файла практически полностью совпадает с файлом для сборки Shiny-приложения, который использовался ранее (**см. раздел 8.4.5.**), за исключением установки других дополнительных пакетов.

Перед сборкой образа необходимо установить актуальные данные подключения к приложению. Так как они будут браться из переменных окружения, которые обычно указываются при запуске, то для их получения необходимо использовать функцию *Sys.getenv("variable_name")*. В файле *app.R* следует описать следующие настройки подключения к базе данных.

```
db <- sd_database(
  host   = Sys.getenv("POSTGRES_HOST"),
  dbname = Sys.getenv("POSTGRES_DB"),
  port   = Sys.getenv("POSTGRES_PORT"),
  user   = Sys.getenv("POSTGRES_USER"),
  table  = Sys.getenv("POSTGRES_TABLE"),
  ignore = FALSE
)
```

Для сборки образа приложения используется команда:

```
docker build -t my-surveydown-app-image .
```

Теперь, когда есть образ приложения, можно описать конфигурацию кластера из двух запускаемых контейнеров - одного приложения и базы данных. Конфигурация Docker Compose описывается в файле *docker-compose.yml*, который можно использовать как для локального запуска на компьютере, так и для запуска на отдельном сервере. Для запуска разрабатываемого приложения данный файл может иметь следующее содержимое.

```
# Описание сервисов, которые будут запущены в кластере.
services:
  # База данных.
  postgres:
    image: postgres:14 # Используемый образ PostgreSQL 14
    container_name: database # Название контейнера для отображения
    hostname: database # Название хоста контейнера внутри сети docker-compose
    restart: unless-stopped  # Контейнер будет перезапускаться в случае сбоя пока
его не остановят
    environment: # Переменные окружения для конфигурации контейнера при запуске
      POSTGRES_DB: dbsurvey  # Название создаваемой базы данных при старте
```

```
      POSTGRES_PASSWORD: dbpassword # Пароль пользователя
      POSTGRES_USER: dbuser # Имя пользователя
    ports:
      - 5432:5432 # Сопоставление внутренних портов контейнера с портами хоста
    volumes:
      - postgres-survey-data:/var/lib/postgresql/data # Папка, где будут храниться
данные Postgres
    networks:
      - survey-networks # Название внутренней сети, в рамках которой будут
находиться сервисы
  # Приложение опросника
  surveyapp:
    image: my-surveydown-app-image
    container_name: surveyapp
    hostname: surveyapp
    restart: unless-stopped
    environment: # Переменные окружения для запуска контейнера
      POSTGRES_HOST: postgres # Адрес для доступа к базе данных внутри сети
      POSTGRES_PORT: 5432 # Внутренний порт контейнера
      POSTGRES_DB: dbsurvey  # Название создаваемой базы данных при старте
      POSTGRES_USER: dbuser # Имя пользователя
      POSTGRES_TABLE: surveytable # Название таблицы для хранения результатов
опроса
      SURVEYDOWN_PASSWORD: dbpassword # Пароль для доступа к базе данных и
администрской странице
    ports:
      - 3838:3838
    depends_on:
      - postgres # Указание, что для приложения требуется запущенная база данных
    networks:
      - survey-networks
# Сети, доступные в кластере
networks:
  survey-networks:
    driver: bridge
# Подключаемые тома данных
volumes:
  postgres-survey-data:
```

Содержимое файла можно разделить на несколько условных блоков:

- *services* - описание сервисов, которые будут запускаться в рамках кластера. Непосредственно для сервисов задаются дополнительные параметры конфигурации, назначение которых описано в комментарии файла.
- *networks* - описание подсетей, которые будут созданы для взаимодействия сервисов. В рамках кластера можно создать несколько подсетей, чтобы сервисы работали независимо друг от друга и могли общаться друг с другом не выходя во внешнюю сеть. В данном случае используется одна подсеть *survey-networks*. Следует отметить, что название сети задается произвольно.
- *volumes* - описание подключенных томов для хранения данных. Поскольку контейнеры внутри себя не сохраняют данные и при повторном создании запускаются "с чистого листа", результаты их работы необходимо где-то сохранить. Для этой цели служат тома. В данном случае создается том для хранения файлов базы данных PostgreSQL. Используется именованный том

postgres-survey-data, который сохраняется непосредственно в папке */var/lib/docker/volumes/*, но при необходимости можно задать и другую папку.

Для загрузки всех нужных образов используется команда:

```
docker compose pull
```

Для запуска всех сервисов, описанных в файле можно использовать команду:

```
docker compose up -d
```

В данном случае атрибут *-d* подсказывает, что после запуска нужно освободить терминал, в этом случае в терминал не будут писаться логи контейнеров. При первом запуске рекомендуется запускать контейнеры по очереди, потому что базе данных Postgres потребуется некоторое время, чтобы создать необходимые файлы. Поэтому лучше сначала выполнить команду:

```
docker compose up -d postgres
```

И лишь затем, через некоторое время (когда в логах контейнера с базой данных появится сообщение *database system is ready to accept connections*) выполнить следующую команду для запуска непосредственно приложения с опросом:

```
docker compose up -d surveyapp
```

Когда все сервисы будут запущены, приложение можно открыть по адресу *http://localhost:3838*. Для теста можно пройти опрос и зайти в админскую страницу по ссылке *http://localhost:3838?admin*. Откроется страница, в которую нужно ввести пароль базы данных (в данном случае “*dbpassword*”). После успешной авторизации откроется страница в которой можно будет приостановить доступность опроса, посмотреть введенные данные или выгрузить результаты.

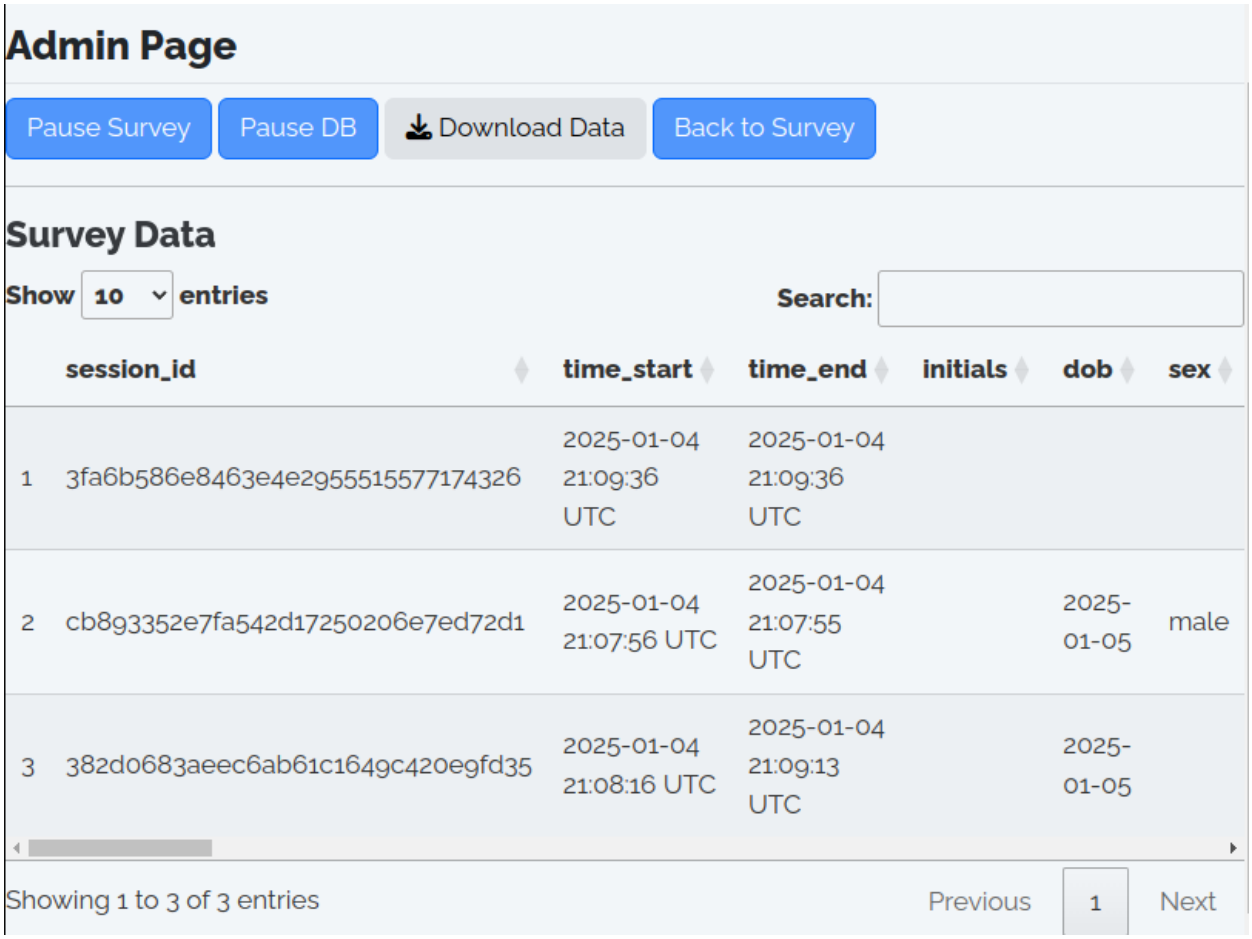


Рисунок 8.207. Страница администратора

Для остановки запущенных сервисов используется команда:

```
docker compose down
```

Таким образом с помощью технологии SurveyDown, можно разрабатывать собственные гибкие опросники с ветвлением, а также обеспечить полный контроль над собираемыми данными.

8.5.2. Система электронной ИРК LibreClinica

В контексте сбора данных для масштабных исследований простых опросных форм становится недостаточно. В таких случаях требуются более комплексные системы, такие как системы управления клиническими данными (CDMS), системы электронного сбора данных (EDC) или электронные индивидуальные регистрационные карты (эИРК/eCRF).

В контексте использования в клинических исследованиях к таким системам предъявляются многочисленные требования, в том числе необходимость авторизации

доступа, аудита изменений, возможности мониторинга и валидации данных и т.д. Обычно подобные программные решения разрабатываются крупными компаниями и обладают значительной стоимостью.

Однако есть и бесплатное open-source решение - LibreClinica (https://www.libreclinica.org) - программное обеспечение с открытым исходным кодом для клинических испытаний, предназначенное для электронного сбора данных (EDC) и управления клиническими данными (CDMS). На самом деле она является преемницей разработанной ранее системы OpenClinica Community и распространяется по лицензии LGPL-3.0, поэтому вся накопленная база знаний по работе с OpenClinica применима и для LibreClinica. В настоящий момент, несмотря на использование несколько устаревших технологий, LibreClinica остается единственным доступным бесплатным средством для организации сбора данных в рамках клинических исследований.

LibreClinica имеет функционал, достаточный для настройки исследования, ввода, мониторинга и извлечения данных. Непосредственно эИРК создаются с помощью файлов Excel, что позволяет сделать их модульными и упрощает повторное использование. Кроме того возможно создание автоматизированных проверок на ввод для исключения некорректных данных. Система обеспечивает контролируемый доступ через учетные записи пользователей и авторизацию, присутствует разделение пользователей по ролям для контроля и проверки вводимых данных. Журнал аудита исследования позволяет просматривать историю изменений введенных данных для каждого субъекта исследования. Предоставляются средства для документирования, сообщения и управления проблемами, связанными с данными в клиническом исследовании. LibreClinica позволяет экспортировать собранные данные клинических исследований в различные форматы (CDISC ODM XML, HTML, Excel, SPSS и Plain Text).

На официальном сайте представлен подробный раздел с документацией (https://www.libreclinica.org/documentation), охватывающий как установку системы, так и руководство для пользователей с различными ролями. Доступна демонстрационная версия системы, данные для подключения к которой доступны по ссылке: https://www.reliatec.de/en/branchen-loesungen/medizin-it/klinische-studien/libreclinica/demo.

Нельзя не упомянуть также ресурс TrialDataSolutions http://trialdatasolutions.com/tds, содержащий достаточно большое количество материалов по работе с системами OpenClinica/LibreClinica.

В рамках данного раздела будет представлен достаточно краткий обзор возможностей системы и рассмотрены сценарии подготовки LibreClinica для запуска на локальном компьютере/удаленном сервере с целью целей разработки эИРК для планируемого исследования. Также будут разобраны основные этапы настройки, ввода и экспорта данных.

Подготовка образа LibreClinica

По умолчанию для работы LibreClinica требуется база данных Postgres и сервер веб приложений Apache Tomcat. Официальная инструкция по установке доступна по ссылке https://www.libreclinica.org/documentation/install.html. В представленном далее примере будет использоваться технология Docker Compose как более универсальное средство. Поэтому прежде всего необходимо подготовить образ с LibreClinica для развертывания.

Для сборки образа LibreClinica будет использоваться следующий Dockerfile.

```
FROM tomcat:9-jdk11
# Создание необходимых папок и настройка перенаправления на LibreClinica
RUN set -eux; \
    mkdir /usr/local/tomcat/webapps/LibreClinica; \
    mkdir /usr/local/tomcat/webapps/ROOT; \
    echo '<html><head><meta http-equiv="refresh" content="0; URL=LibreClinica/"
/></head></html>' \
        > /usr/local/tomcat/webapps/ROOT/index.html;
# Скачивание установочного пакета с LibreClinica и распаковка в папку приложений
Apache Tomcat.
RUN apt-get update -qq; \
    apt-get -y --no-install-recommends install unzip; \
    curl -o /tmp/LibreClinica.war
https://www.libreclinica.org/downloads/LibreClinica-web-1.3.0.war; \
    unzip -q /tmp/LibreClinica.war -d /usr/local/tomcat/webapps/LibreClinica/; \
    rm /tmp/LibreClinica.war
# Добавление тома для хранения данных и логов.
VOLUME \
    /usr/local/tomcat/libreclinica.data \
    /usr/local/tomcat/logs
# Добавление файлов конфигурации.
COPY \
    ./config/ \
    /usr/local/tomcat/libreclinica.config/
# Добавление перевода на русский язык.
COPY \
    ./i18n/translation/ \
    /usr/local/tomcat/webapps/LibreClinica/WEB-
INF/classes/org/akaza/openclinica/i18n/
# Добавление виджета календаря на русском языке
COPY \
    ./i18n/calendar/ \
    /usr/local/tomcat/webapps/LibreClinica/includes/new_cal/lang/
```

В качестве базового образа используется *tomcat:9-jdk11* - образ Debian с уже предустановленным Apache Tomcat 9 и Java Development Kit 11. Далее скачивается установочный файл LibreClinica-web-1.3.0.war, который по сути является zip-архивом и распаковывается в папку приложений Apache Tomcat. Затем копируются файлы конфигурации (самая важная часть). В данных файлах определяются параметры подключения к базе данных.

В файле *datainfo.properties* присутствует блок со следующими параметрами подключения:

```
dbType=postgres
dbUser=clinica
dbPass=clinica
db=libreclinica
dbPort=5432
dbHost=database
```

Представленные параметры служат примером и могут быть изменены в соответствии с требованиями. Важно предварительно настроить запускаемый сервер Postgres для работы с указанными параметрами доступа.

Поскольку LibreClinica по умолчанию имеет англоязычный интерфейс, на заключительном этапе сборки образа производится ее русификация. Для этого загружаются соответствующие словари и виджет календаря.

Для сборки данного образа необходимо выполнить следующую команду:

```
docker build -t my-libreclinica-image .
```

Запуск LibreClinica

После того как образ собран, можно перейти к файлу *docker-compose.yml* для запуска кластера. В данном файле для базы данных важно прописать в переменных окружения те параметры, которые были указаны в файле *datainfo.properties*.

```
# Описание сервисов, которые будут запущены в кластере.
services:
  # Приложение LibreClinica.
  libreclinica:
    image: docker.io/library/my-libreclinica-image
    container_name: libreclinica # Название контейнера для отображения
    hostname: libreclinica # Название хоста контейнера внутри сети docker-compose
    restart: unless-stopped  # Контейнер будет перезапускаться в случае сбоя пока
его не остановят
    ports:
      - 8080:8080
    volumes:
      - libreclinica-data:/usr/local/tomcat/libreclinica.data # Данные
LibreClinica
      - libreclinica-logs:/usr/local/tomcat/logs # Логи LibreClinica
    depends_on:
      - database
    networks:
      - libreclinica-network # Название внутренней сети, в рамках которой будут
находиться сервисы
  # База данных.
  database:
    image: postgres:14 # Используемый образ PostgreSQL 14
    container_name: database # Название контейнера для отображения
```

```
    hostname: database # Название хоста контейнера внутри сети docker-compose
    restart: unless-stopped  # Контейнер будет перезапускаться в случае сбоя пока
его не остановят
    environment:
      POSTGRES_PASSWORD: clinica
      POSTGRES_USER: clinica
      POSTGRES_DB: libreclinica
    volumes:
      - postgres-libreclinica-data:/var/lib/postgresql/data # Папка, где будут
храниться данные Postgres
    networks:
      - libreclinica-network # Название внутренней сети, в рамках которой будут
находиться сервисы
# Сети, доступные в кластере.
networks:
  libreclinica-network:
    driver: bridge
# Подключаемые хранилища данных.
volumes:
  postgres-libreclinica-data:
  libreclinica-data:
  libreclinica-logs:
```

Для запуска кластера необходимо ввести следующую команду:

```
docker compose up -d
```

Следует дождаться, когда в логах контейнера *libreclinica* появится сообщение *org.apache.catalina.startup.Catalina.start Server startup in [73899] milliseconds*. Длительность первого старта может отличаться, т.к. при запуске приложение заполняет базу данных необходимыми таблицами и информацией, что влияет на скорость работы.

После запуска приложения в браузере по адресу http://localhost:8080 будет доступна страница входа в LibreClinica (**рисунок 8.208**).

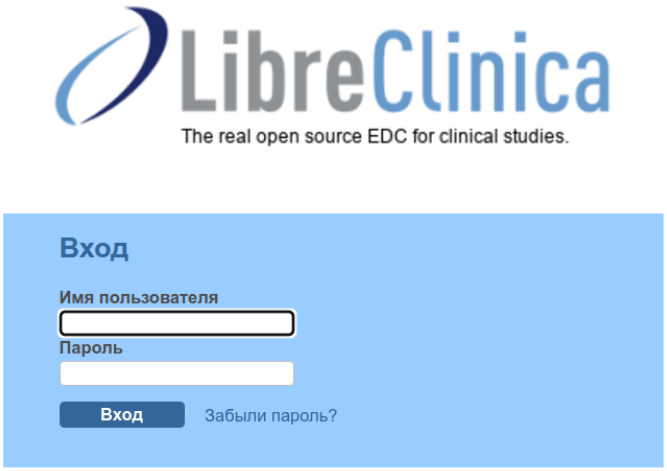


Рисунок 8.208. Страница входа в LibreClinica

Настройка исследования в LibreClinica

По умолчанию логином для входа корневого (главного) пользователя служит *root*, паролем - *12345678*. После первого входа будет предложено сменить пароль. В дальнейшем в настройках системы можно будет настроить время, в течении которого пароль будет считаться актуальным. Пользователь *root* считается главным администратором системы. После установки нового пароля откроется стартовая страница с тестовым исследованием по умолчанию.

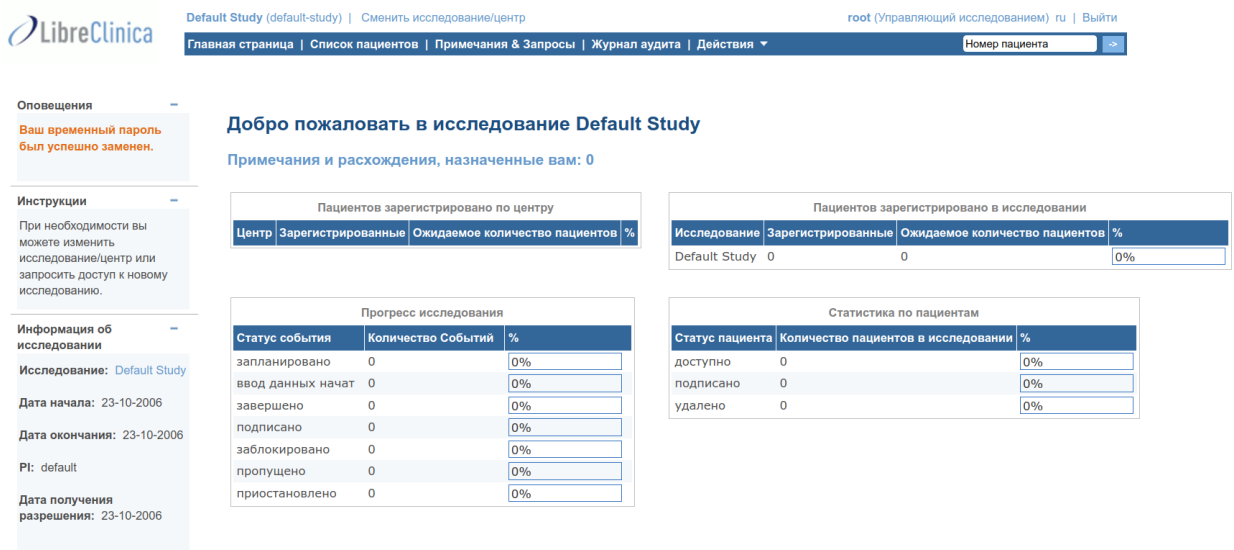


Рисунок 8.209. Страница с тестовым исследованием

Для создания нового исследования необходимо выбрать пункт *Действия -> Администратор: Исследования*. Откроется страница зарегистрированных в системе исследований. В рамках одной установки LibreClinica можно одновременно создавать несколько параллельных исследований с различными параметрами. Данные исследования будут работать независимо друг от друга.

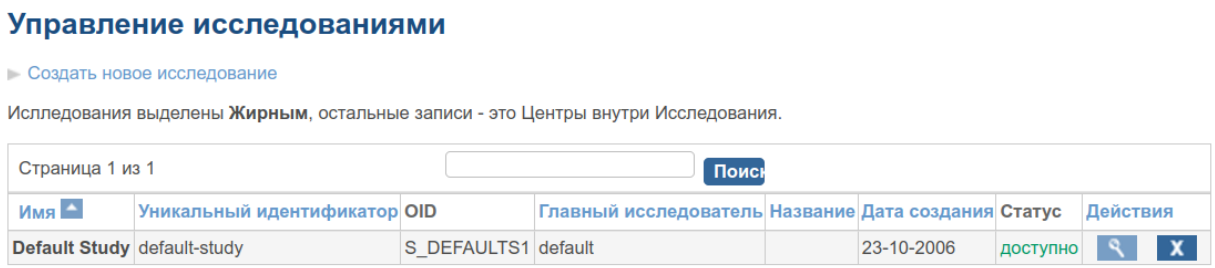


Рисунок 8.210. Страница с доступными исследованиями

Для создания нового исследования (**рисунок 8.210**) необходимо нажать кнопку “Создать новое исследование”. После этого потребуется внести некоторую регистрационную информацию про исследование. Рекомендуется параметры “Уникальный номер протокола” и “Краткое название” вносить латинскими буквами.

Рекомендуется чтобы краткое название не превышало 8 символов - так будет проще впоследствии именовать связанные с этим исследованием сущности, добавляя перед именем краткое название: центра, события, ИРК, и т.д. В остальном все введенные на этой странице данные носят справочно-информационный характер и не влияют на работу системы.

Создать новое исследование

Enter the Study and Protocol Information requested below to create the study. The data element definitions the website for descriptions of all fields.

РАЗДЕЛ А: Описание исследования

* указывает обязательные для заполнения поля.

Уникальный номер протокола: TEST-DEMO-STUDY *

Краткое название: TST *

Название центра: TSTMAIN

Вторичные IDs: (через запятую)

Главный исследователь: TestInvestigator *

Тип протокола: Интервенционный Наблюдательное *

Login E-Mail Notification: Disabled Enabled *

Project Manager Contact E-mail:

Резюме: Информация о проводимом исследовании *

Подробное описание:

Спонсор: n_a *

Сотрудники: (через запятую)

Please select a user who will later edit or update the Study.

Выбрать пользователя : root (Root User)

Сохранить Отменить

Рисунок 8.211. Страница создания нового исследования

После заполнения всех обязательных полей и нажатия кнопки "Сохранить", пользователь, обладающий правами администратора, перенаправляется на страницу "Управление исследованиями". В таблице исследований отображается вновь созданное исследование.

Для последующей работы с созданным исследованием необходимо перейти к нему. Для этого пользователь должен выполнить следующие действия:

1. В верхней левой части страницы выбрать опцию "Сменить исследование/центр".
2. Выбрать вновь созданное исследование.
3. Нажать кнопку "Изменить исследование" (**рисунок 8.212**).

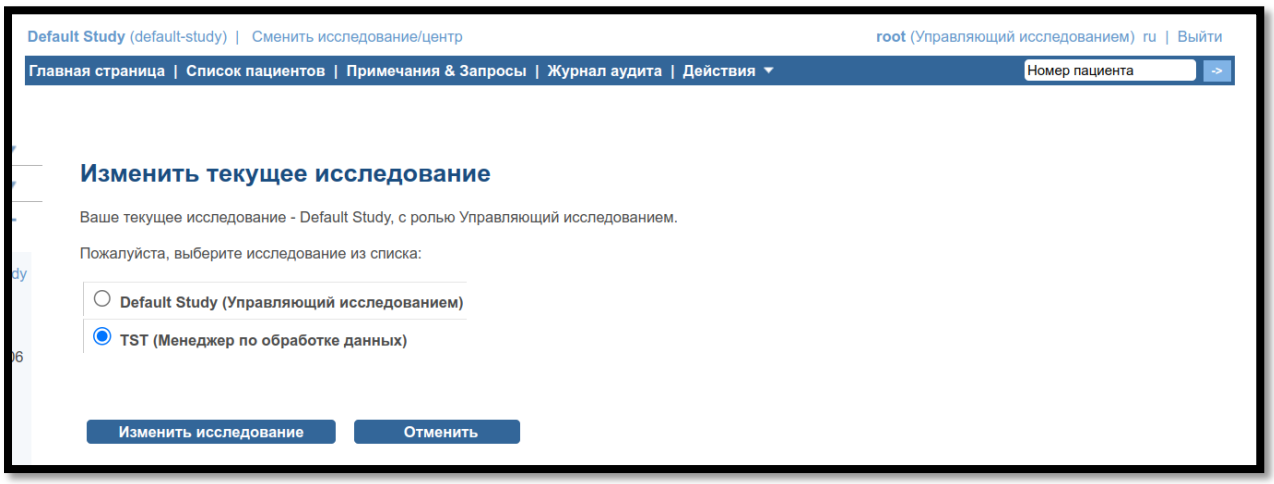


Рисунок 8.212. Страница изменения (выбора) исследования

После этого администратор попадает на страницу управления исследованием. Сам процесс настройки состоит из семи шагов, которые нужно пройти последовательно (**рисунок 8.213**).

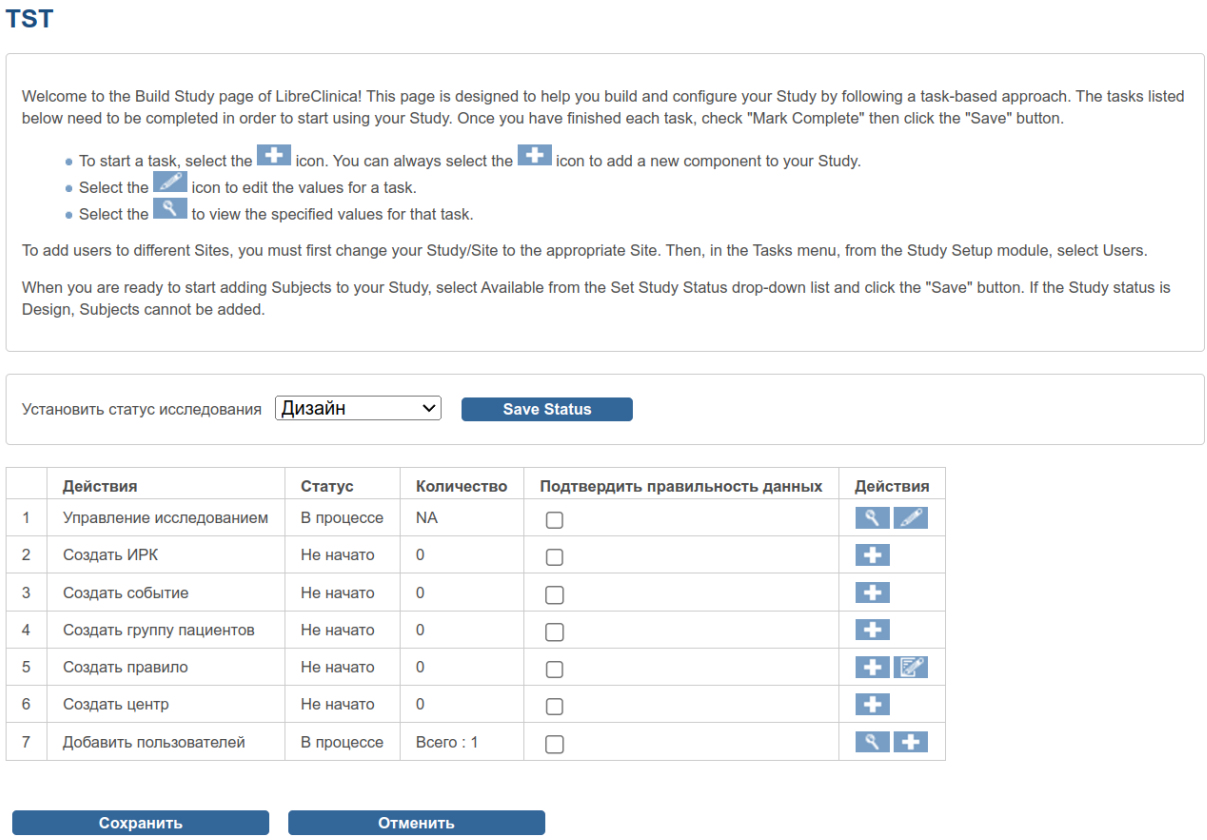


Рисунок 8.213. Страница настройки исследования (7 шагов)

Следует обратить внимание, что созданное исследование в данный момент находится в статусе “Дизайн”. Это означает, что оно находится в процессе настройки: в нем можно менять параметры, добавлять новые ИРК, центры и так далее, но нельзя вводить данные пациентов. Когда процесс настройки будет закончен, исследование нужно перевести в статус “Доступно”. Тогда оно будет считаться доступным для пользовательского ввода данных. Когда процесс ввода данных в исследование необходимо завершить - устанавливается статус “Заморожено”. После этого все введенные данные переходят в режим “только чтение для зарегистрированных пользователей”, за исключением пользователей-администраторов исследования, которым разрешена работа с примечаниями и уточнениями. Статус “Заблокировано” блокирует изменение данных уже для всех пользователей без исключения. Администратор системы может удалить исследование, в таком случае у него будет статус “Удалено”. Однако важно отметить, что на самом деле практически никакие введенные данные не удаляются из системы, а лишь помечаются удаленными.

Теперь следует перейти к шагам по подготовке исследования к работе. Полное и исчерпывающее руководство представлено в официальной документации к OpenClinica (https://docs.openclinica.com/3-1/study-setup). Далее будут рассмотрены ключевые моменты по работе с исследованием.

Главной единицей сбора информации в рамках исследования считается “Объект исследования/Пациент (Subject)”. У каждого “Объекта” исследования должен быть уникальный в рамках всего исследования идентификационный номер и дата начала участия в исследовании. С объектом исследования могут происходить заранее определенные “События” (Study Events), в ходе которых вносятся данные в одну или несколько ИРК (индивидуальных регистрационных карт; CRF). Эту структуру элементов необходимо определить в ходе настройки исследования.

Первый этап заключается в заполнении регистрационных данных исследования и конфигурации особенностей создания и оформления событий, происходящих с пациентом (Subject), который является основной единицей сбора данных. Наиболее значимым в этом плане является раздел “Конфигурация параметров исследования”.

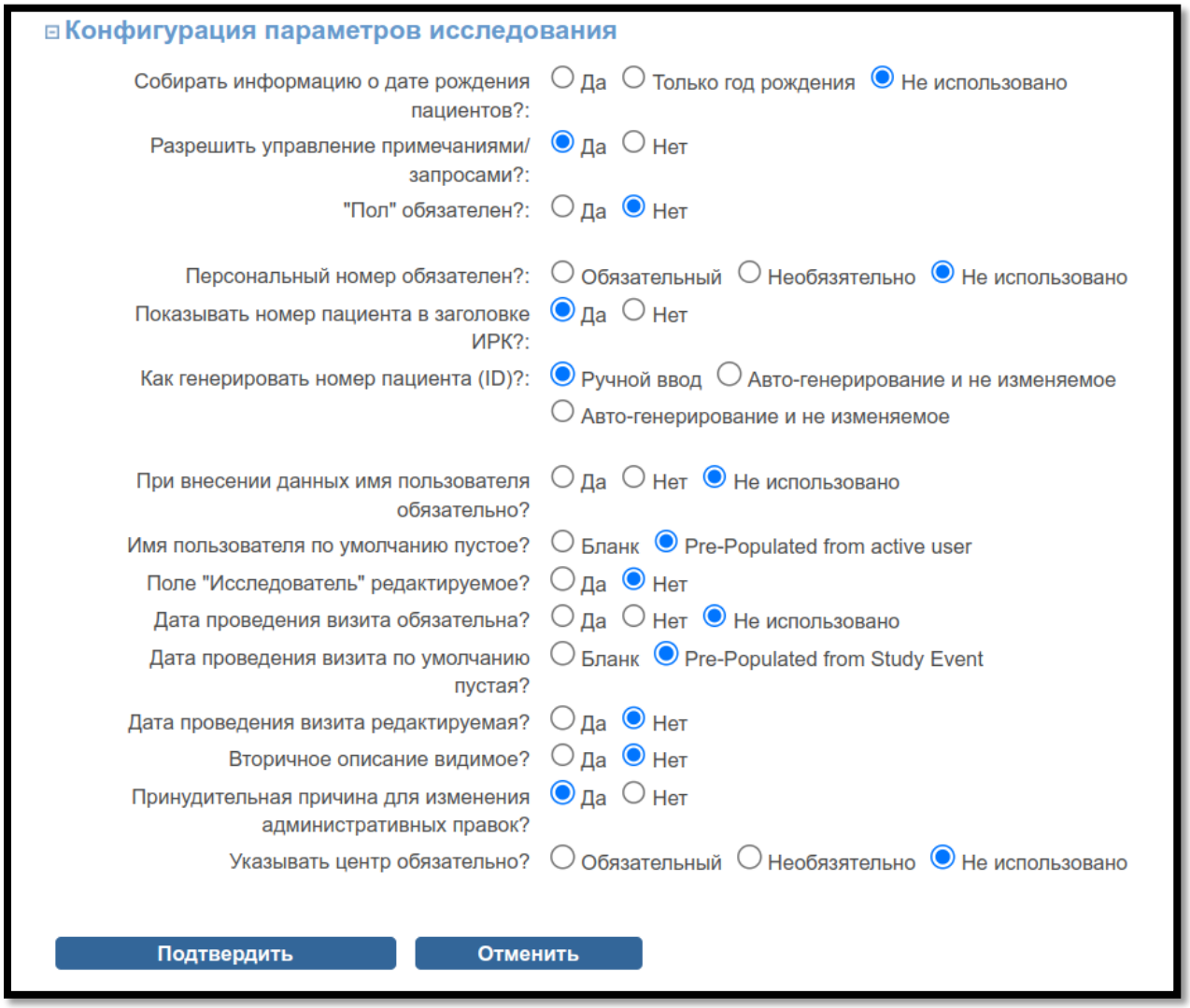
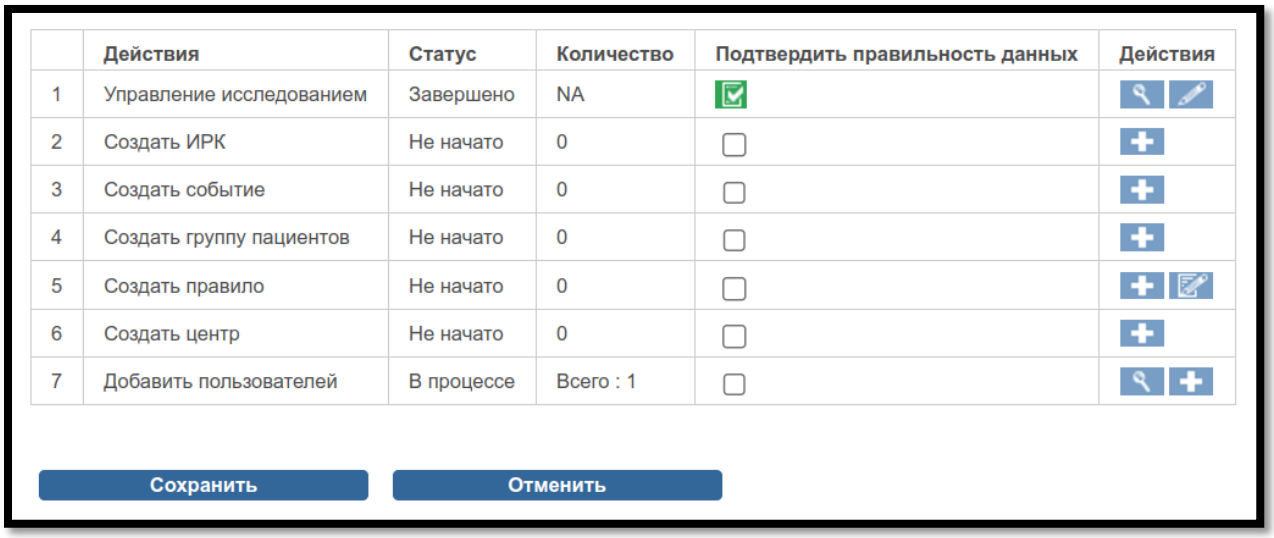


Рисунок 8.214. Страница конфигурации параметров исследования

На рассматриваемой странице пользователь может конфигурировать следующие параметры исследования:

- Обязательность заполнения полей при создании пациента.
- Использование уточняющих примечаний.
- Обязательность заполнения дат для событий.

Рекомендуется отключать обязательность заполнения пола и даты рождения пациента, поскольку эти данные целесообразнее собирать непосредственно в рамках исследования (ИРК) вместе с другими полями. Кроме того, целесообразно установить ручной ввод для поля "Номер пациента" (Subject ID), поскольку в нем обычно содержится кодировка центров, которая может потребовать редактирования в ходе исследования. После внесения необходимых изменений пользователь должен нажать кнопку "Подтвердить". Откроется страница настроек исследования, где можно подтвердить окончание работы с первым этапом, установив флажок в первом разделе (“Управление исследованием”) и сохранив изменения (**рисунок 8.215**).



	Действия	Статус	Количество	Подтвердить правильность данных	Действия
1	Управление исследованием	Завершено	NA	☑	
2	Создать ИРК	Не начато	0	☐	
3	Создать событие	Не начато	0	☐	
4	Создать группу пациентов	Не начато	0	☐	
5	Создать правило	Не начато	0	☐	
6	Создать центр	Не начато	0	☐	
7	Добавить пользователей	В процессе	Всего : 1	☐	

Сохранить Отменить

Рисунок 8.215. Страница конфигурации параметров исследования с завершенным первым шагом

Теперь необходимо загрузить используемые ИРК. Каждая ИРК представляет собой файл Excel с расширением .xls, в котором описывается структура полей и экранных форм, которые должны отобразиться на странице. Чистый шаблон документа для заполнения можно скачать прямо на странице.

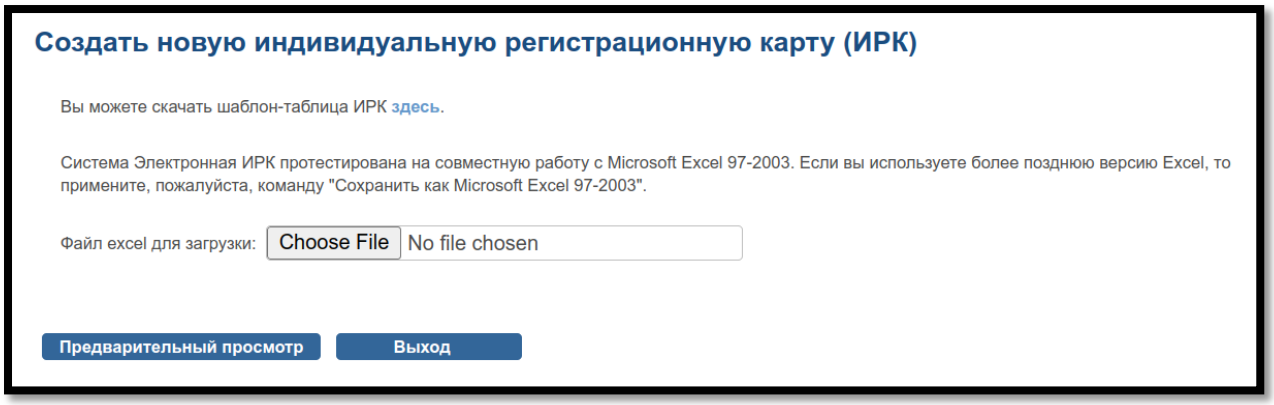


Рисунок 8.216. Страница создания (загрузки) ИРК

Полное руководство по созданию ИРК доступно на официальном сайте OpenClinica (https://docs.openclinica.com/3-1/study-setup/study-setup-build-study-create-case-report-forms-crfs). Кроме того, на сайте также доступен также набор готовых ИРК и правил для них, в которых представлены примеры всех возможностей при оформлении и размещении компонентов OpenClinica CRF Features Kitchen Sink (https://www.openclinica.com/blog/openclinica-crf-features-everything-but-the-kitchen-sink/). Данные примеры можно использовать как шаблоны для создания новых ИРК.

Далее рассмотрены основные принципы построения ИРК на примере Kitchen Sink. Файл с расширением .xls состоит из четырех листов. Условно можно сказать, что они содержат в себе иерархическое описание ИРК сверху вниз.

Лист *CRF* описывает название и версию ИРК. Каждая ИРК кроме названия может иметь несколько версий, которые могут отличаться стилем оформления и набором полей. Благодаря этому уже в ходе исследования можно вносить изменения, загружать обновленные версии карт и переносить прежние данные. Механизм версий позволяет также создавать схожие, но немного отличающиеся ИРК для разных центров или групп пациентов, или создавать одинаковые формы, но на нескольких языках.

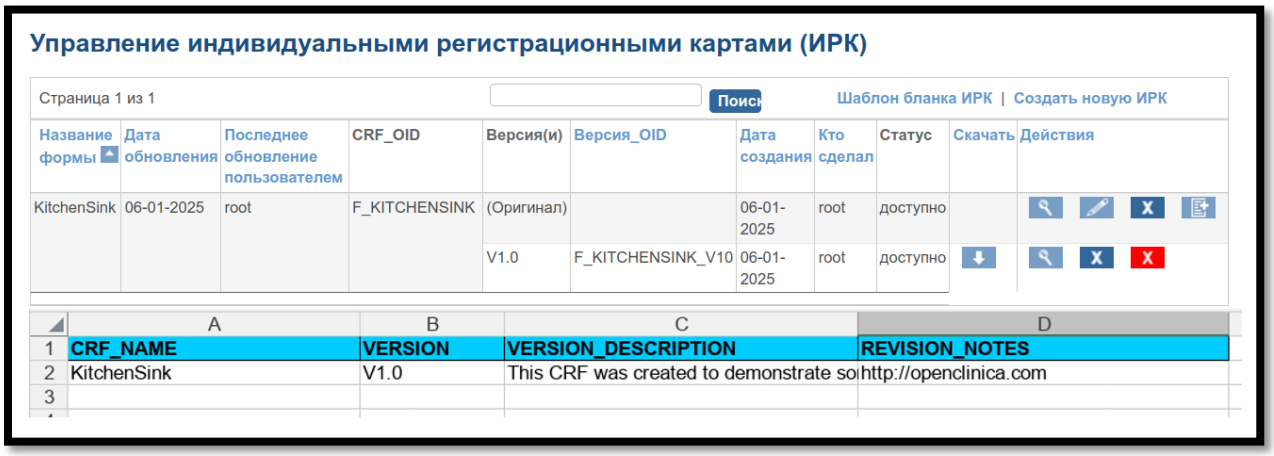


Рисунок 8.217. Лист CRF

На листе *Sections* представлен перечень вкладок, на которые разбита ИРК. Несмотря на то, что такая возможность присутствует, рекомендуется ограничиваться одной вкладкой на странице, потому что пользователи весьма часто пропускают вкладки и большое количество вкладок затрудняет ввод.

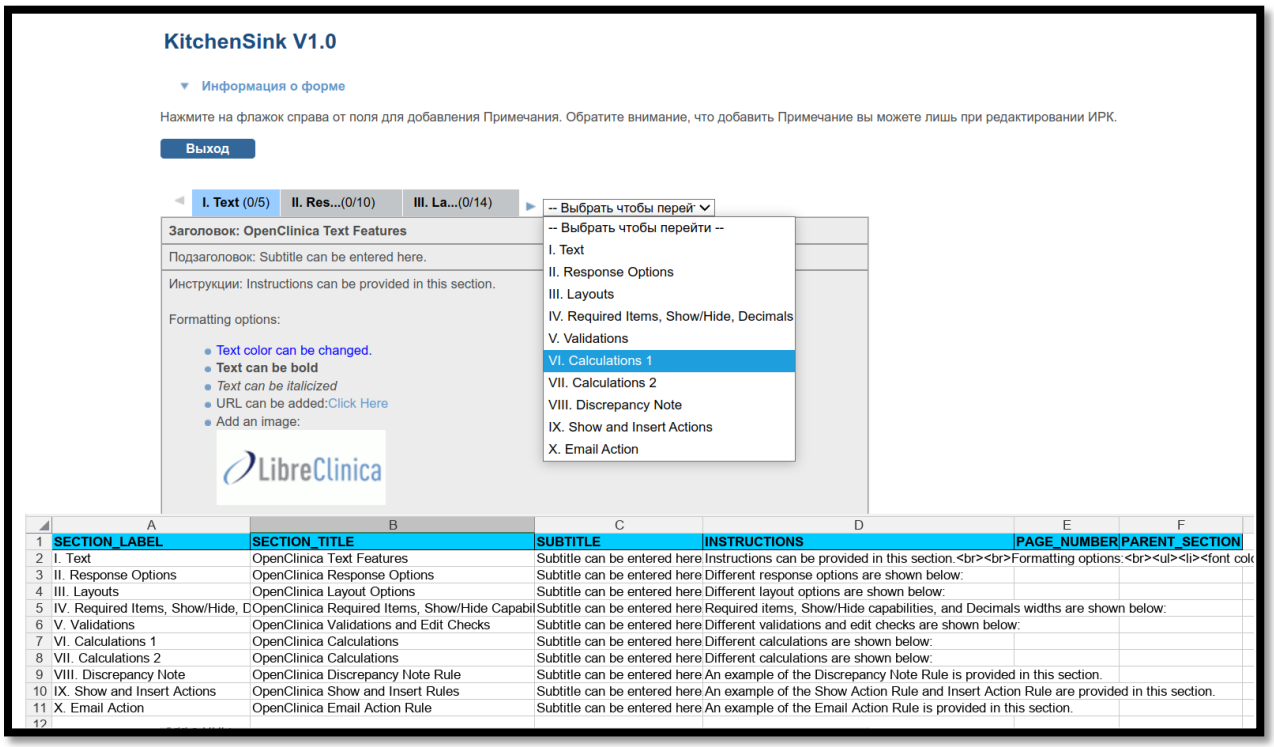


Рисунок 8.218. Лист Sections

На листе *Groups* представлены группы полей. На вкладках поля собираются в смысловые группы, чтобы их было потом проще анализировать. На одной вкладке может быть одна и более групп полей. Группы могут быть двух типов: NON-REPEATING - когда поля расположены одно под другим в виде списка; GRID - когда поля формируют строку таблицы. Для имен групп рекомендуется использовать короткие наименования, не превышающие 8 символов, например PAT - для данных пациентов, LAB - для лабораторных анализов и т.д.

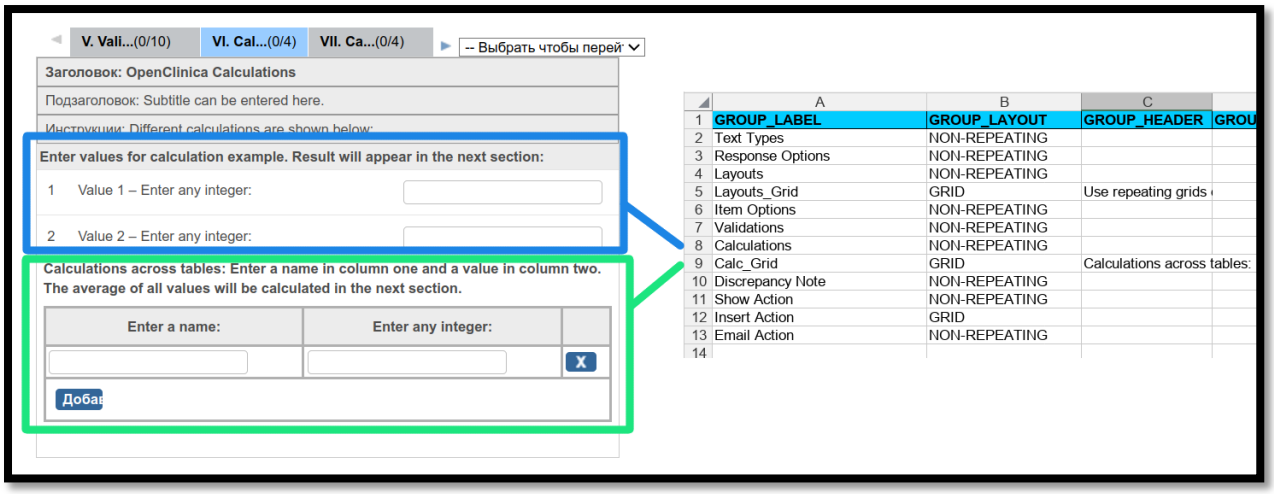
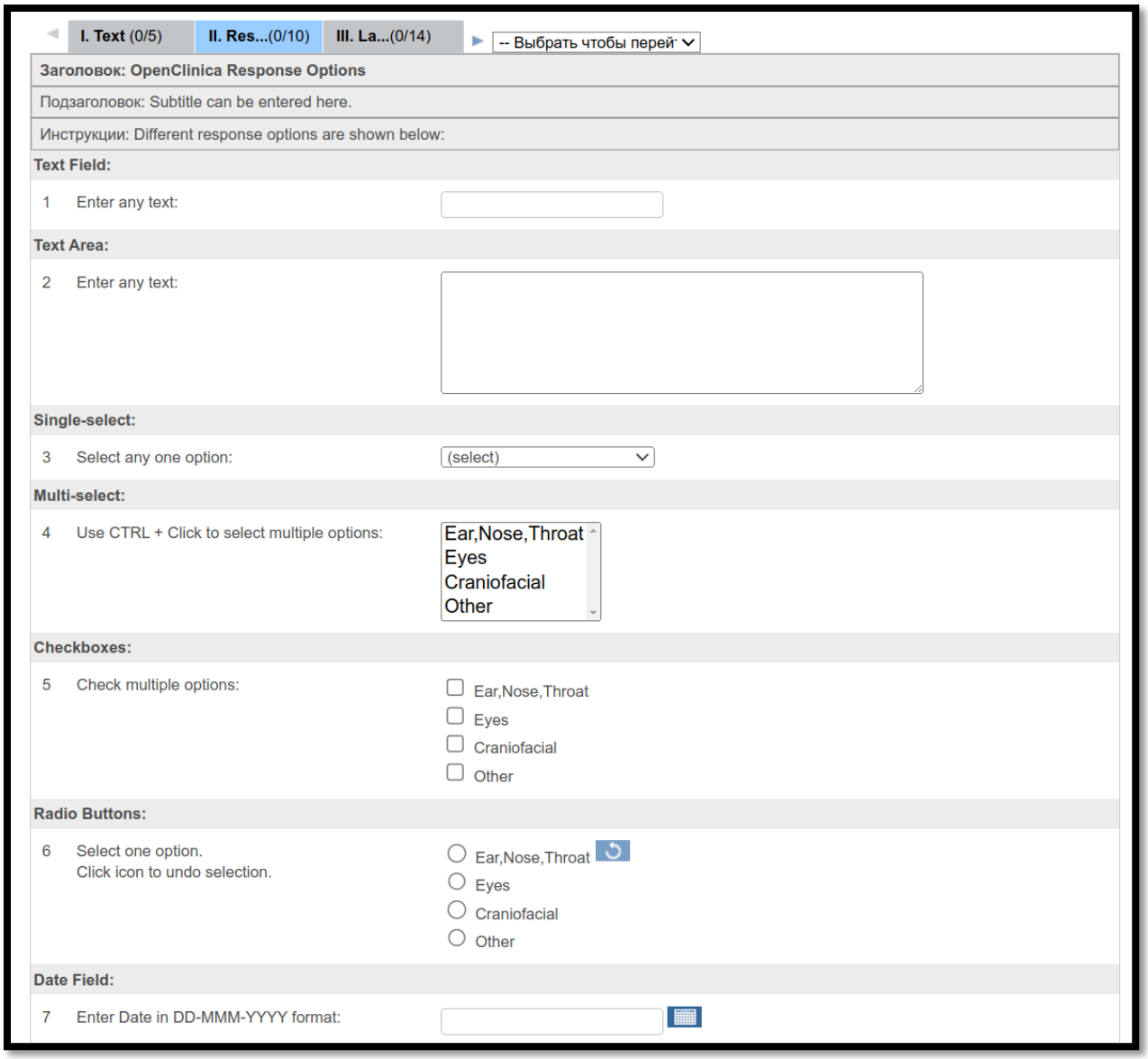
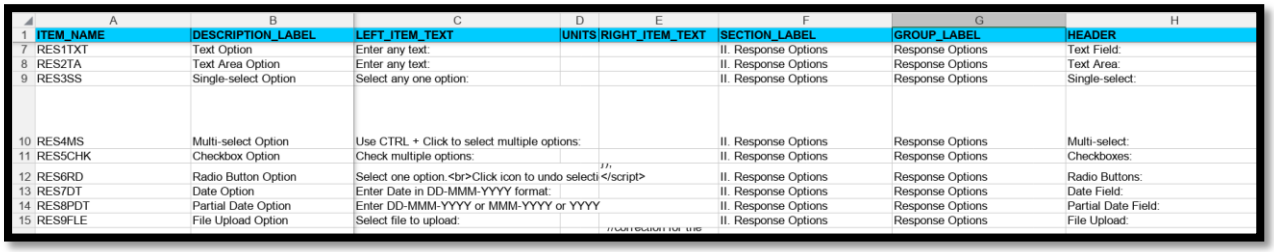


Рисунок 8.219. Лист Groups

Лист *Items* содержит описания полей. Для каждого поля можно задать его имя, настройки визуального отображения и принадлежность к определенной группе и вкладке. Особое внимание стоит обратить на поле ITEM_NAME - это название поля, как оно будет сохраняться в базе данных. Оно может содержать только латинские буквы, цифры и символ нижнего подчеркивания. Кроме того ITEM_NAME каждого поля должно быть уникально в рамках ИРК. Поскольку имя поля чувствительно к регистру, рекомендуется использовать преимущественно верхний регистр и использовать имена не превышающие 32 символов, так как именно эти имена будут использоваться в последующей выгрузке. Также рекомендуется начинать имя переменной с кода группы, к которой она относится - это упростит дальнейшую обработку данных при экспорте. Например, для данных пациента использовать PAT_SEX - для пола, PAT_BIRTHDATE - для даты рождения и т.д.





	A	B	C	D	E	F	G	H
1	ITEM_NAME	DESCRIPTION_LABEL	LEFT_ITEM_TEXT	UNITS	RIGHT_ITEM_TEXT	SECTION_LABEL	GROUP_LABEL	HEADER
7	RES1TXT	Text Option	Enter any text:			II. Response Options	Response Options	Text Field:
8	RES2TA	Text Area Option	Enter any text:			II. Response Options	Response Options	Text Area:
9	RES3SS	Single-select Option	Select any one option:			II. Response Options	Response Options	Single-select:
10	RES4MS	Multi-select Option	Use CTRL + Click to select multiple options:			II. Response Options	Response Options	Multi-select:
11	RES5CHK	Checkbox Option	Check multiple options:			II. Response Options	Response Options	Checkboxes:
12	RES6RD	Radio Button Option	Select one option. Click icon to undo selecti		//; </script>	II. Response Options	Response Options	Radio Buttons:
13	RES7DT	Date Option	Enter Date in DD-MMM-YYYY format:			II. Response Options	Response Options	Date Field:
14	RES8PDT	Partial Date Option	Enter DD-MMM-YYYY or MMM-YYYY or YYYY			II. Response Options	Response Options	Partial Date Field:
15	RES9FLE	File Upload Option	Select file to upload:			II. Response Options	Response Options	File Upload:

Рисунок 8.220. Лист Items

Раздел *Instructions* не относится непосредственно к функционалу ИРК, а содержит инструкции по заполнению и справочную информацию по полям предыдущих листов.

Несмотря на то, что в примере Kitchen Sink используется несколько вкладок, такой способ организации форм не выглядит оптимальным. Гораздо более предпочтительным выглядит подход, когда вместо одной максимально возможной

ИРК создаются несколько “атомарных” ИРК. Например, если в рамках визита пациента собираются демографические данные, критерии включения/исключения и лабораторные исследования, то предпочтительнее вместо одной ИРК с тремя вкладками сделать три отдельных ИРК. Это позволит переиспользовать существующие разделы в других событиях, вместо создания новых. Например, когда исследование включает в себя три визита пациента и на каждом требуется ввод данных лабораторных исследований, можно создать одну ИРК и использовать ее в качестве шаблона на каждом визите.

С примерами таких ИРК можно ознакомиться по ссылке в репозитории.

	Действия	Статус	Количество	Подтвердить правильность данных	Действия
1	Управление исследованием	Завершено	NA	☑	
2	Создать ИРК	Завершено	14	☑	
3	Создать событие	Не начато	0	☐	
4	Создать группу пациентов	Не начато	0	☐	
5	Создать правило	Не начато	0	☐	
6	Создать центр	Не начато	0	☐	
7	Добавить пользователей	В процессе	Всего : 1	☐	

Рисунок 8.221. Страница конфигурации параметров исследования с завершенным вторым шагом

Третий этап настройки исследования заключается в создании событий, которые могут происходить с пациентом.

При создании события необходимо указать несколько параметров, которые определяют его тип и поведение. Рекомендуется использовать максимально короткое, но в тоже время понятное имя события, т.к. каждое событие при вводе на интерфейсе будет создавать отдельный столбец в таблице пациентов. При большом количестве событий такая таблица может стать слишком широкой и неудобной для работы. Описание события может раскрывать подробности события и содержать более длинный текст.

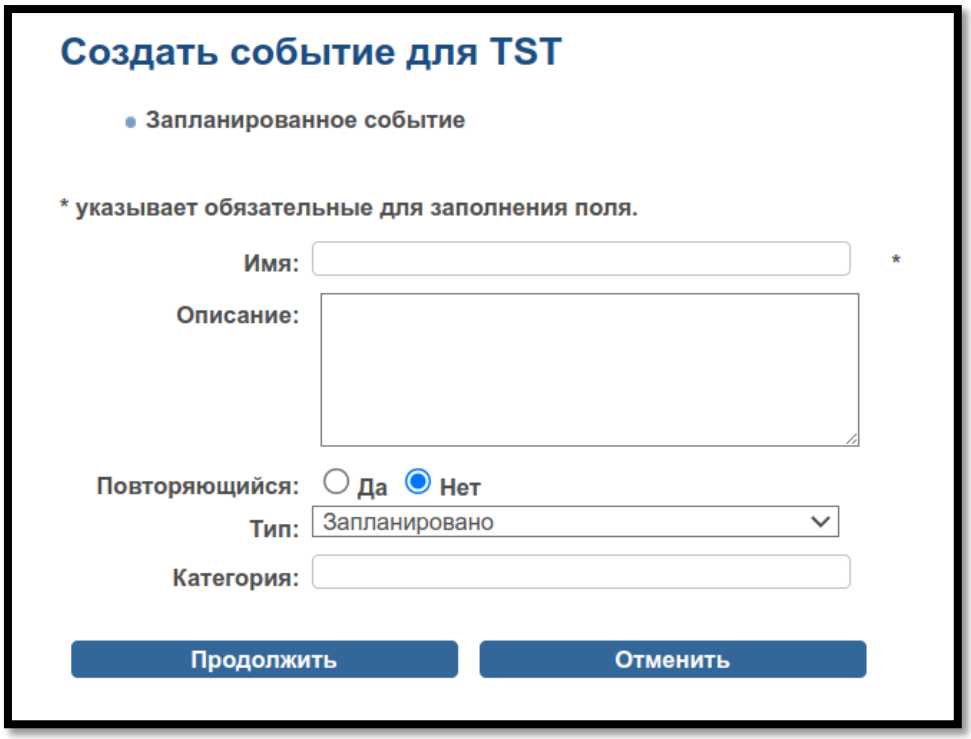


Рисунок 8.222. Страница создания события

События могут быть нескольких типов:

- Запланированное - событие, которое обязательно должно произойти в рамках исследования. Без заполнения данных по нему, запись пациента не будет считаться полной. Например, это может быть визит включения пациента в исследование.
- Незапланированное - событие, которое может произойти в ходе исследования, а может и не произойти. Например, случай возникновения серьезного нежелательного явления.
- Общее - событие, которое может содержать в себе какие-либо дополнительные или необязательные данные. В отличие от запланированных и незапланированных событий, этот тип события может не относиться непосредственно к какому-либо моменту времени. Например, это может быть анамнез пациента или комментарии исследователя.

Отдельным пунктом можно задать повторение события:

- Повторяющиеся события - это события одного типа, которые могут происходить неоднократно в ходе исследования. Предполагаются, что в рамках этого события будут заполняться одни и те же ИРК, а меняться будет только дата возникновения события. В этом случае в таблице пациентов рядом со значком события на этапе ввода будет отображаться количество введенных событий. К ним можно отнести промежуточные визиты пациента или лабораторные исследования.

- Неповторяющиеся события - это события, которые могут возникнуть только один раз в ходе исследования. Например, визиты включения пациента в исследование или наоборот, визит завершения.

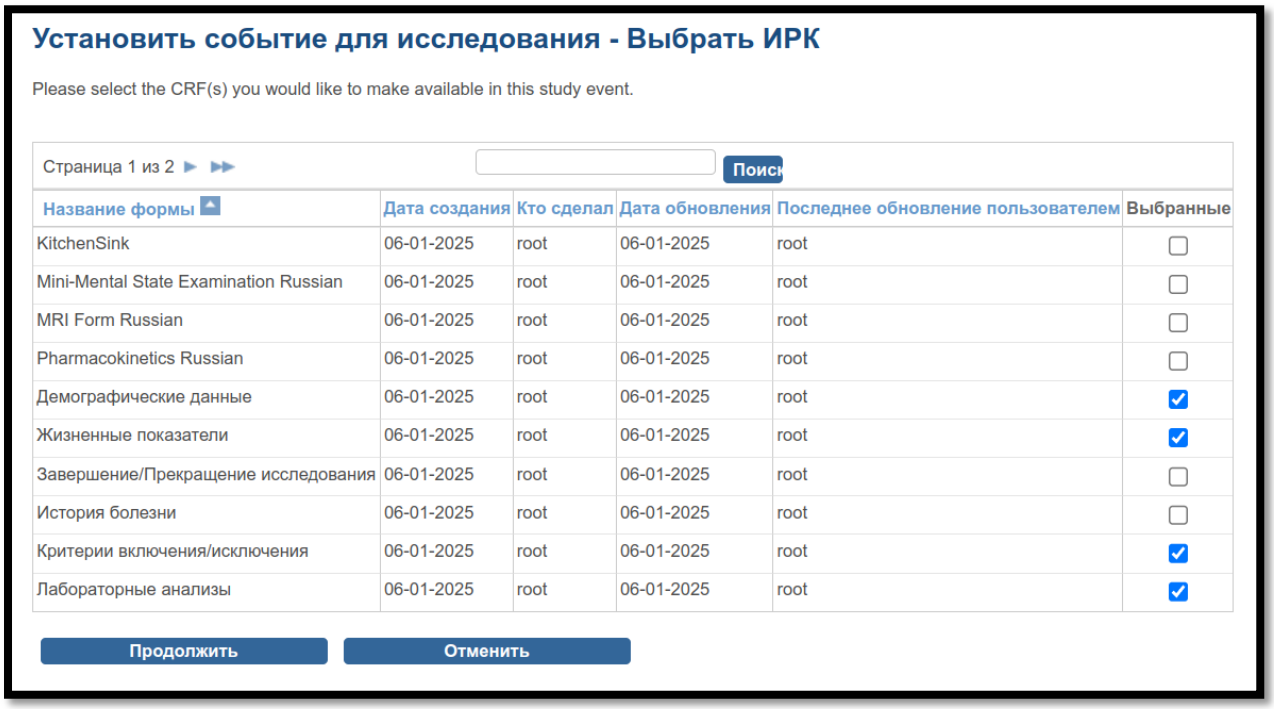
Категория позволяет визуально объединить события в блоки и не несет отдельной смысловой нагрузки.

В рамках представленного тестового исследования будут созданы следующие события:

- Визит включения: запланированное и неповторяющееся;
- Визит завершения: запланированное и неповторяющееся;
- Лабораторное исследование: запланированное и повторяющееся;
- Серьезное нежелательное явление: незапланированное и повторяющееся;
- История болезни: общее и неповторяющееся.

После создания события необходимо назначить для него ИРК, которые нужно заполнить в рамках события - одну или несколько. Например, в рамках события “Визит включения” будут заполняться следующие ИРК (**рисунок 8.223**):

- Критерии включения/исключения;
- Демографические данные;
- Жизненные показатели;
- Обследование;
- Лабораторные анализы.



Название формы	Дата создания	Кто сделал	Дата обновления	Последнее обновление пользователем	Выбранные
KitchenSink	06-01-2025	root	06-01-2025	root	☐
Mini-Mental State Examination Russian	06-01-2025	root	06-01-2025	root	☐
MRI Form Russian	06-01-2025	root	06-01-2025	root	☐
Pharmacokinetics Russian	06-01-2025	root	06-01-2025	root	☐
Демографические данные	06-01-2025	root	06-01-2025	root	☑
Жизненные показатели	06-01-2025	root	06-01-2025	root	☑
Завершение/Прекращение исследования	06-01-2025	root	06-01-2025	root	☐
История болезни	06-01-2025	root	06-01-2025	root	☐
Критерии включения/исключения	06-01-2025	root	06-01-2025	root	☑
Лабораторные анализы	06-01-2025	root	06-01-2025	root	☑

Рисунок 8.223. Страница назначения ИРК на событие

Следующим шагом необходимо настроить поведение ИРК в рамках события: выбрать используемую версию, если их несколько, отметить обязательность заполнения формы. Стоит отметить, что можно выбрать несколько версий ИРК для одного события и на этапе ввода оператор сможет выбрать какую именно он будет заполнять. Данный подход может быть полезен при использовании одинаковых форм на разных языках. Обязательность заполнения формы влияет на статус события - оно не будет считаться завершенным, пока не будут введены данные во все обязательные формы.

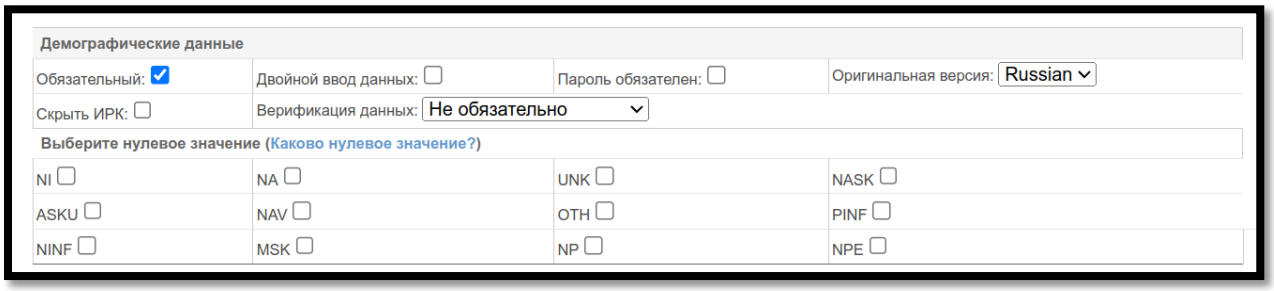


Рисунок 8.224. Настройка поведения ИРК

Пункт “Скрыть ИРК” позволяет убрать форму при вводе карты оператором, но показать ее администратору исследования, чтобы он мог вносить какие-либо свои данные или комментарии.

Пункт “Двойной ввод данных” включает дополнительный механизм контроля правильности ввода данных, когда одну и ту же ИРК вводят отдельно два различных оператора и после второго ввода система проверяет введенные ИРК на наличие расхождений и в случае возникновения ошибок предлагает исправить их.

Пункт “Верификация данных” предполагает в процессе исследования верификацию данных мониторами и активирует режим работы системы в котором данные не считаются готовыми к обработке, пока их не проверит отдельный пользователь с ролью монитор.

Группа переключателей “Выберите нулевое значение” позволяет выбрать одно или несколько значений, которые будут добавлены к каждому из полей ввода в рамках ИРК, где можно будет отметить факт отсутствия отметок, недоступности данных и т.д.

После этого нужно подтвердить создание события и перейти к следующему. Формы можно переиспользовать между различными событиями, поэтому для события “Лабораторные исследования” можно снова выбрать ИРК “Жизненные показатели” и “Лабораторные анализы”.

Таким образом, последовательно создаются все события и назначаются нужные ИРК для каждого из них. Впоследствии можно изменить порядок расположения событий в таблице списка пациентов, открыв список событий в настройках исследования (**рисунок 8.225**).

Manage All Event Definitions in Study TST

Found 5 Definitions Поиск Скрыть

Порядок	Имя	OID	Повторяющийся	Тип	Категория	Популяционный	Дата создания	Дата обновления	ИРК	Оригинальная версия	Действия
	Визит включения	SE_4584	Нет	Запланировано	Визит	Нет	06-01-2025 (root)	06-01-2025 (root)	Демографические данные	Russian	
									Жизненные показатели	Russian	
									Критерии включения/исключения	Russian	
									Лабораторные анализы	Russian	
	Лабораторное исследование	SE_2077	Да	Запланировано	Визит	Нет	06-01-2025 (root)	06-01-2025 (root)	Жизненные показатели	Russian	
									Лабораторные анализы	Russian	
									Регистрация нежелательных явлений	Russian	
	Визит завершения	SE_1080	Нет	Запланировано	Визит	Нет	06-01-2025 (root)	06-01-2025 (root)	Завершение/Прекращение исследования	Rus	
	История болезни	SE_7424	Нет	Общее		Нет	06-01-2025 (root)	06-01-2025 (root)	История болезни	Russian	
									Сопутствующие препараты	Russian	
	Серьезное нежелательное явление	SE_4020	Да	Незапланированные		Нет	06-01-2025 (root)	06-01-2025 (root)	Серьезное нежелательное явление	Russian	

Рисунок 8.225. События и назначенные ИРК

Порядок следования ИРК в рамках события также можно изменить, открыв событие в режиме просмотра.

Просмотреть события

Имя:	Визит включения
OID:	SE_4584
Описание:	
Повторяющийся:	Нет
Тип:	scheduled
Категория:	Визит

ИРК

Click the "up/down" arrow icons in the "Order" column in the following table to change the order of all CRFs.

Порядок	Имя	Обязательный	Двойной ввод данных	Пароль обязателен	Оригинальная версия	Скрыть ИРК	Нулевое значение	Верификация данных	Статус	Действия
	Демографические данные	Да	Нет	Нет	Russian	No		Не обязательно	доступно	
	Обследование	Да	Нет	Нет	Russian	No		Не обязательно	доступно	
	Жизненные показатели	Да	Нет	Нет	Russian	No		Не обязательно	доступно	
	Критерии включения/ исключения	Да	Нет	Нет	Russian	No		Не обязательно	доступно	
	Лабораторные анализы	Да	Нет	Нет	Russian	No		Не обязательно	доступно	

Вернуться к списку событий

Рисунок 8.226. Настройка порядка следования ИРК в рамках события

На данном этапе при необходимости можно завершить настройку исследования и переключить его в режим “Доступно”. В этом случае у исследователей появится возможность вносить данные.

Следующие этапы настройки исследования (этапы 4-7) используются по мере необходимости в соответствии с требованиями конкретного исследования.

	Действия	Статус	Количество	Подтвердить правильность данных	Действия
1	Управление исследованием	Завершено	NA	☑	
2	Создать ИРК	Завершено	14	☑	
3	Создать событие	Завершено	5	☑	
4	Создать группу пациентов	Не начато	0	☐	
5	Создать правило	Не начато	0	☐	
6	Создать центр	Не начато	0	☐	
7	Добавить пользователей	В процессе	Всего : 1	☐	

Рисунок 8.227. Общий вид списка этапов по настройке исследования

Настройка групп пациентов позволяет на этапе создания новой записи пациента определить его в какую-либо заранее определенную категорию - например группа рандомизации.

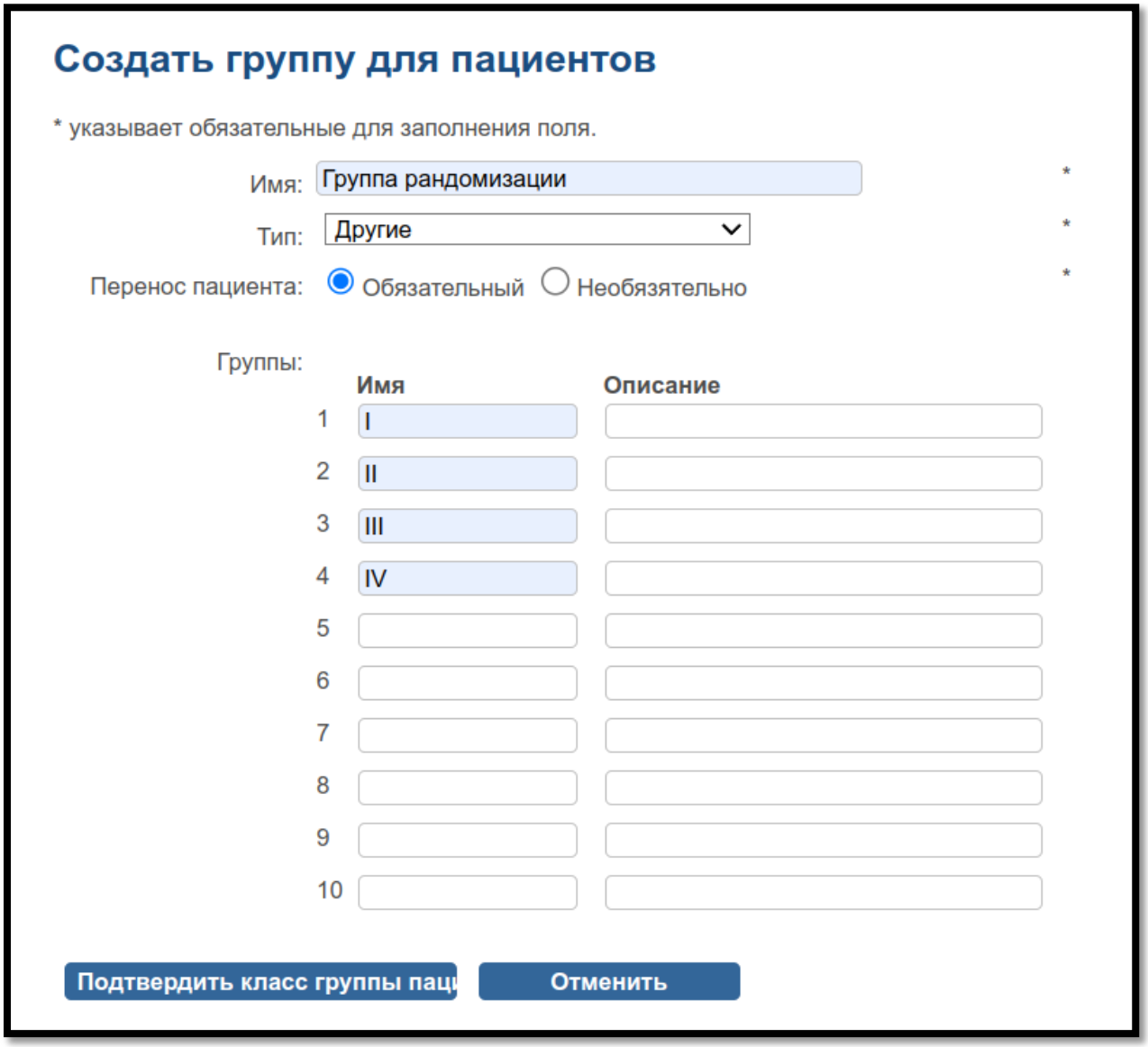


Рисунок 8.228. Настройка групп пациентов

Таких групп может быть несколько, они могут быть как обязательными, так и необязательными для заполнения.

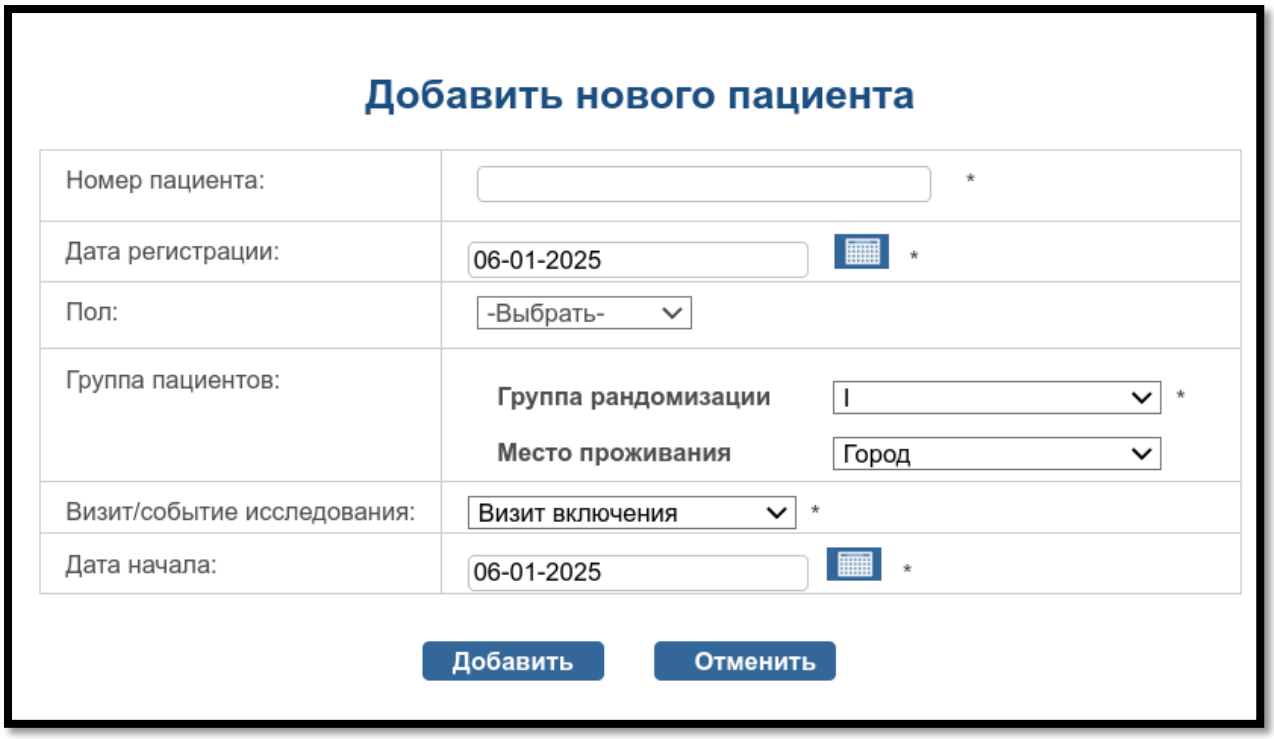


Рисунок 8.229. Страница добавления нового пациента

Создание правил позволяет описывать автоматизированные действия при создании событий и вводе данных в ИРК: создание новых событий, показ и скрытие элементов, создание примечания, вставка заранее определенных данных, отправка email. Такие правила описываются в формате XML (**см. раздел 2.5.6.**) и загружаются в систему с привязкой к конкретным типам событий и полям ИРК. Работа с правилами требует отдельного погружения и выходит за рамки настоящего издания. С созданием и настройкой правил можно ознакомиться в официальной документации OpenClinica (https://docs.openclinica.com/3-1/rules).

Для многоцентровых исследований будет полезен пункт создания центров и заведения отдельных пользователей для каждого из них. Таким образом можно задать разграничение зон видимости, чтобы пользователи могли работать только с доступными им данными, а не со списком пациентов всего исследования. Это позволит на этапе ввода привязать каждого пациента к определенному центру, но при этом сохраняется возможность переносить пациентов между центрами. Для каждого такого центра можно задать отдельные настройки для событий и используемых версий ИРК. Кроме того для каждого пользователя можно определить роли в рамках центра - оператор ввода данных, исследователь, монитор или администратор.

В рамках данного примера настройка исследования будет считаться завершенной. Таким образом, для исследования можно установить статус “Доступно” и перейти к рассмотрению процесса ввода данных.

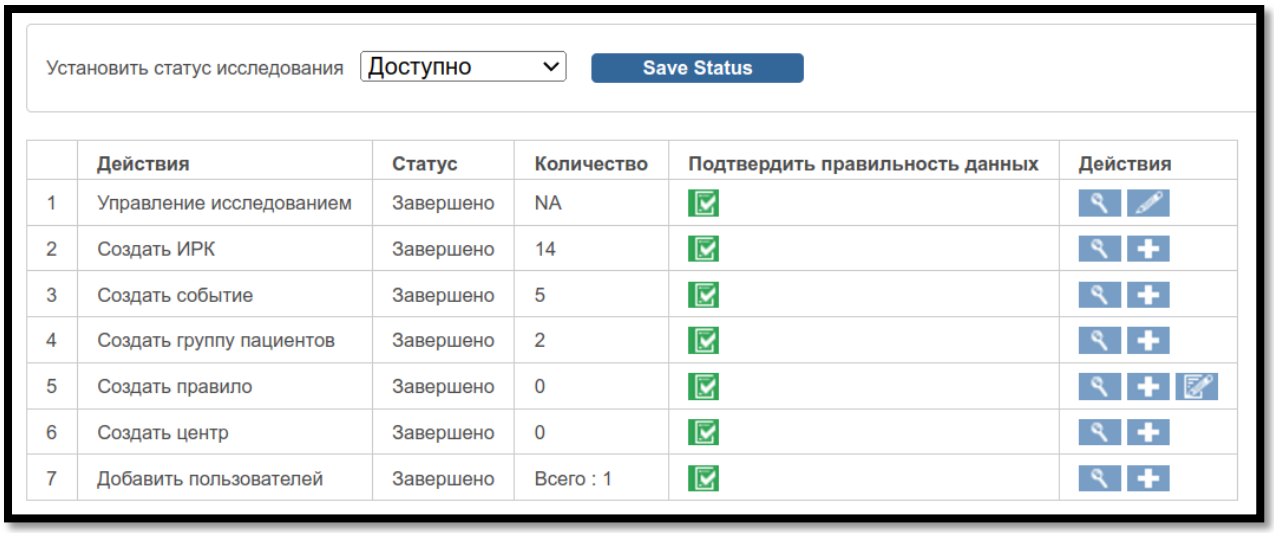


Рисунок 8.230. Установление статуса “Доступно” для исследования

Ввод данных в LibreClinica

Ввод данных пациентов осуществляется на странице “Список пациентов”. После перевода исследования в статус “Доступно” на нем появляется кнопка “Добавить нового пациента”.

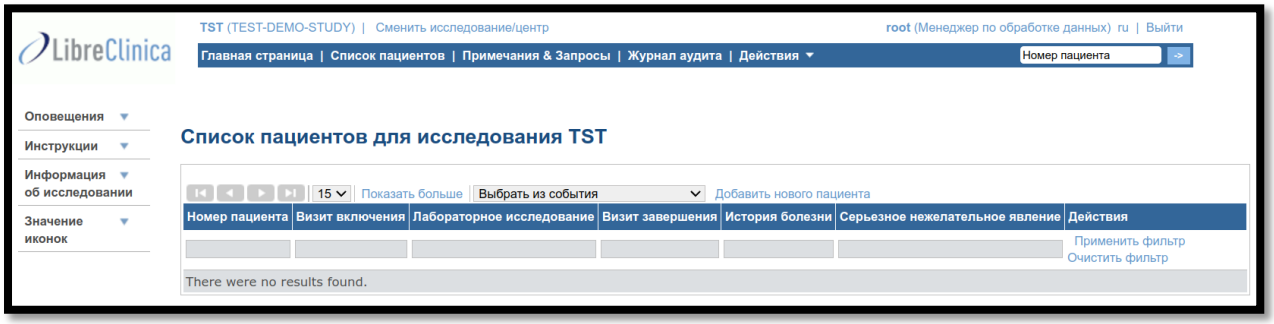


Рисунок 8.231. Страница со списком пациентов для ввода данных

Создание пациента таким образом позволяет сразу выбрать события, которые будут созданы с указанием даты начала события. Помимо даты начала, каждое событие также может иметь дату завершения, которая может быть установлена позже. Однако в большинстве случаев такая установка не требуется.

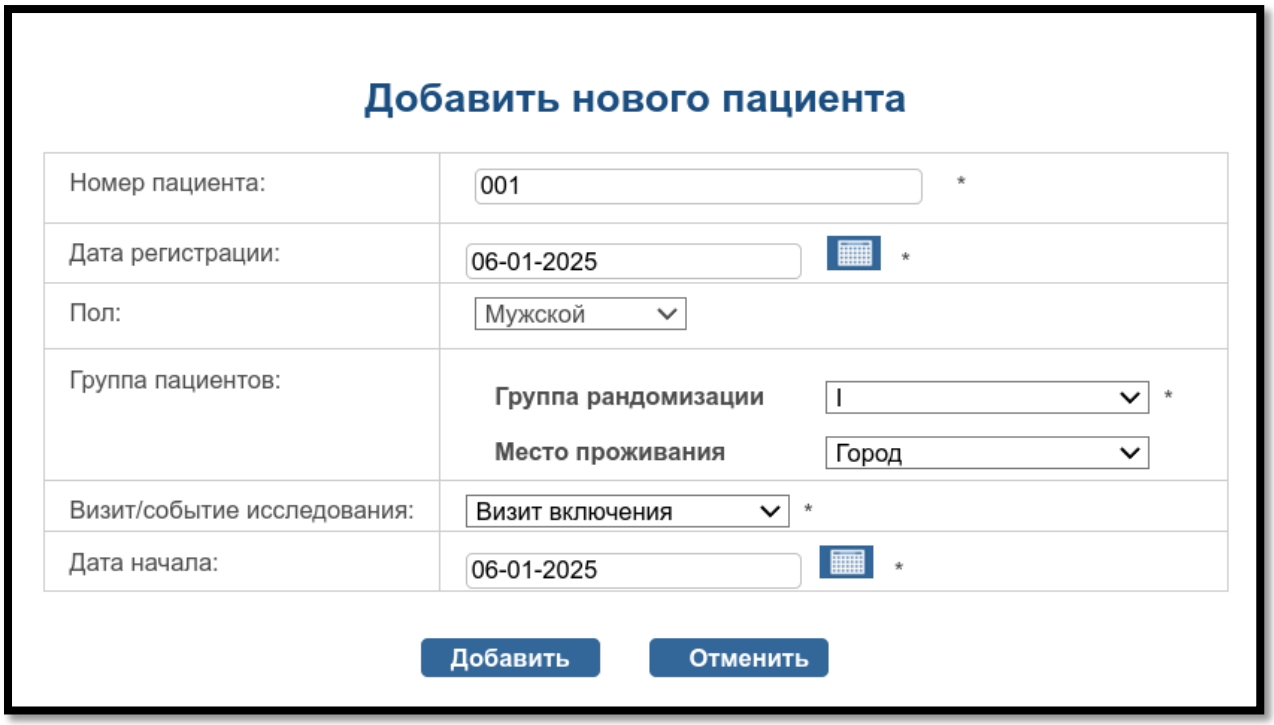


Рисунок 8.232. Страница для добавления нового пациента

После завершения добавления пациента отображается страница просмотра пациента со списком зарегистрированных событий (**рисунок 8.233**). В данном примере создано только одно событие, но их количество может увеличиваться в дальнейшем. Пока ввод данных не начинался, рядом с каждой ИРК стоит синий значок (расшифровку значков можно посмотреть в левой части страницы).

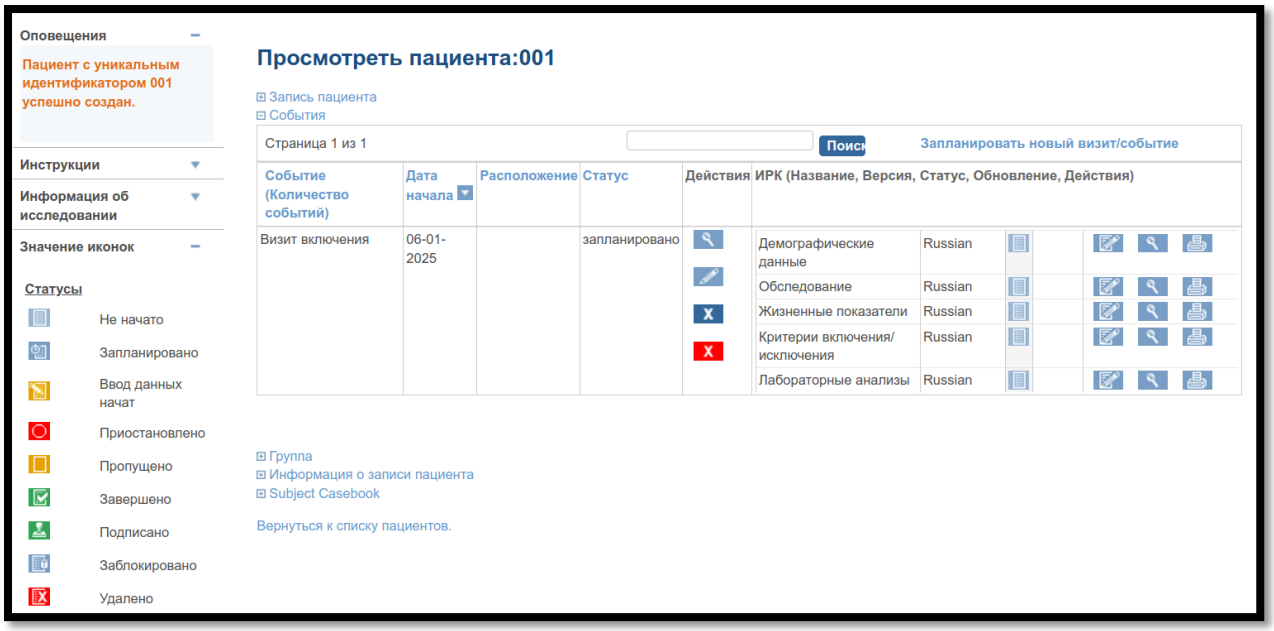


Рисунок 8.233. Страница пациента с зарегистрированным событием “Визит включения”

Теперь необходимо последовательно заполнить ИРК для данного события (“Демографические данные”, “Обследование”, “Жизненные показатели”, “Критерии включения/исключения”, “Лабораторные анализы”) с помощью кнопки ввод данных с изображением листа и карандаша.

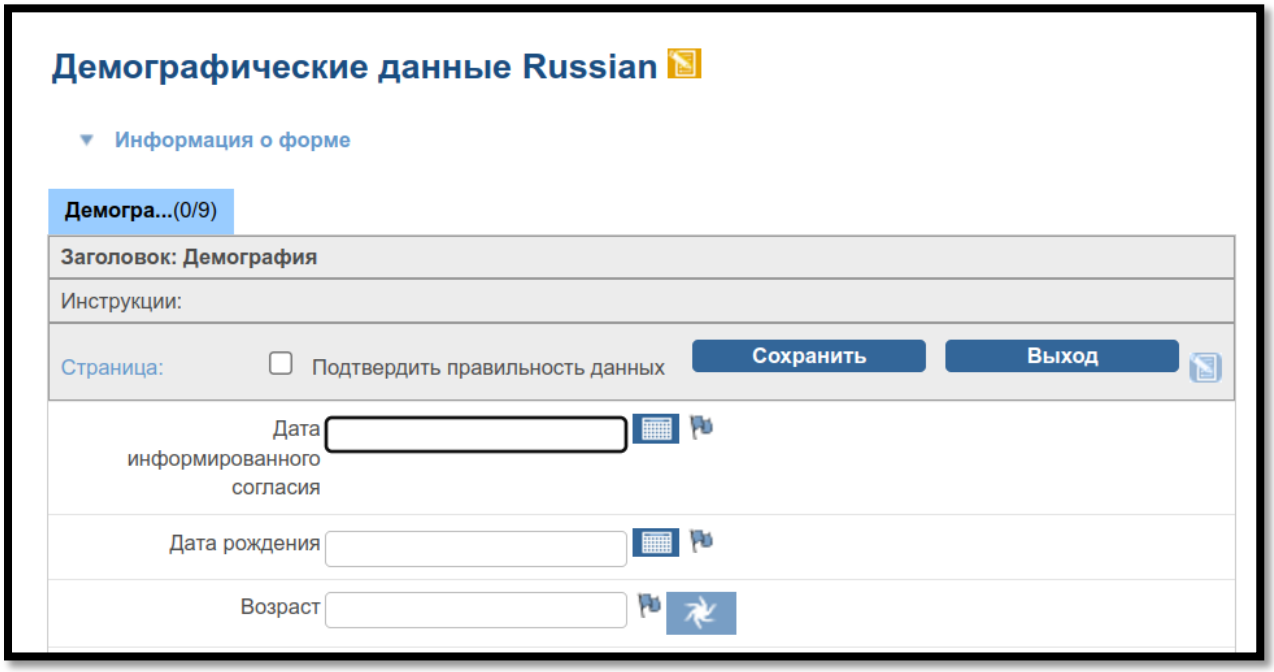


Рисунок 8.234. Заполнение ИРК “Демографические данные”

После заполнения всех полей необходимо нажать кнопку “Сохранить”, и если все проверки валидации данных прошли успешно, то откроется страница события, а статус соответствующей ИРК станет оранжевым - “Ввод данных начат” (**рисунок 8.235**).

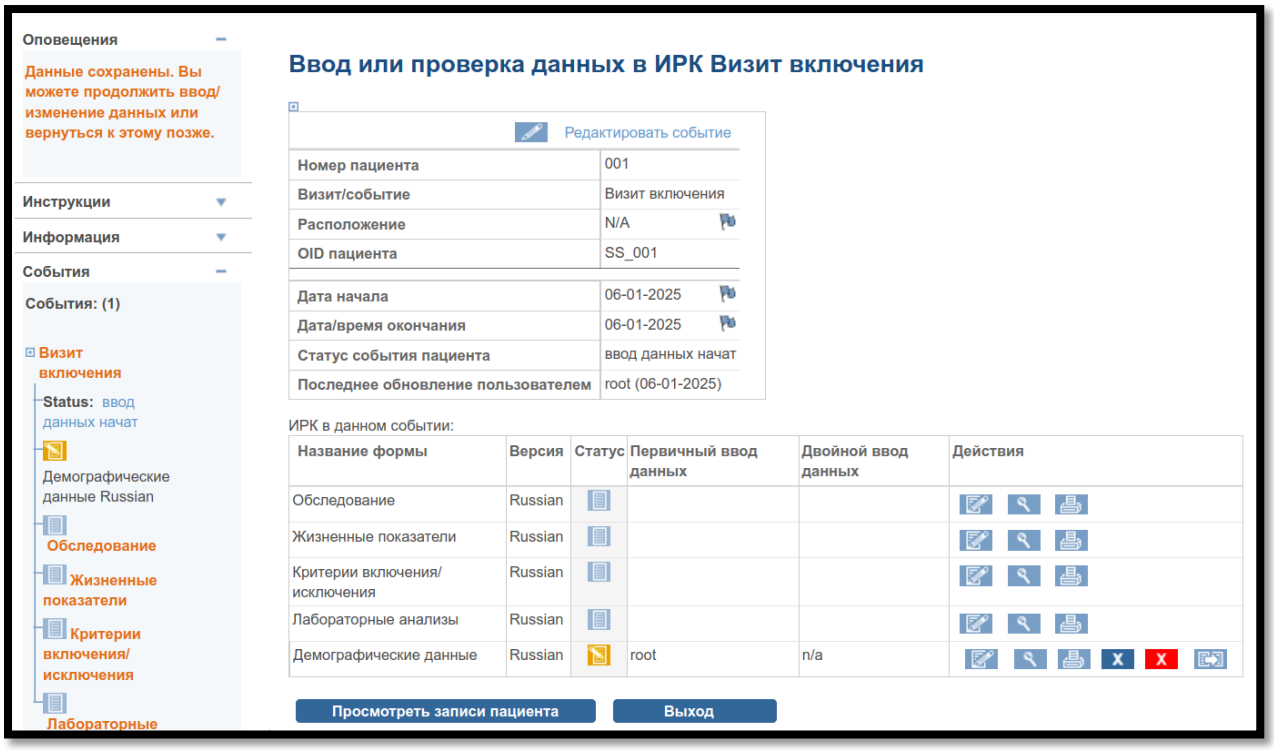


Рисунок 8.235. Страница события “Визит включения”,
статус “Ввод данных начат” для ИРК “Демографические данные”

“Оранжевый” статус означает, что пользователь открыл данную ИРК, внес какие-либо данные, но еще не закончил работу с ней. Возможно были внесены не все поля, либо подразумевается дальнейшее изменение введенной информации. Таким образом, данная форма еще не считается завершенной, она не может быть валидирована и следовательно не будет участвовать в дальнейшей обработке.

Чтобы данная ИРК считалась завершенной, перед сохранением нужно отметить галочку “Подтвердить правильность данных” и уже потом нажать кнопку “Сохранить”. Сохраненные таким образом ИРК отмечаются зелеными значками с галочкой - “Ввод данных завершен” (**рисунок 8.236**). Следует отметить, что после сохранения таким образом ИРК считается завершенной, и все последующие ее модификации требуют указания причин изменения и создания отдельных примечаний для исследователя-монитора. Кроме того, если на момент редактирования данная ИРК была валидирована, то ее утвержденный статус будет сброшен, и потребуется повторная валидация.

Чтобы событие считалось полностью заполненным все ИРК в рамках данного события должны получить “зеленый” статус. До этого момента событие будет иметь оранжевый статус - “Ввод данных начат”.

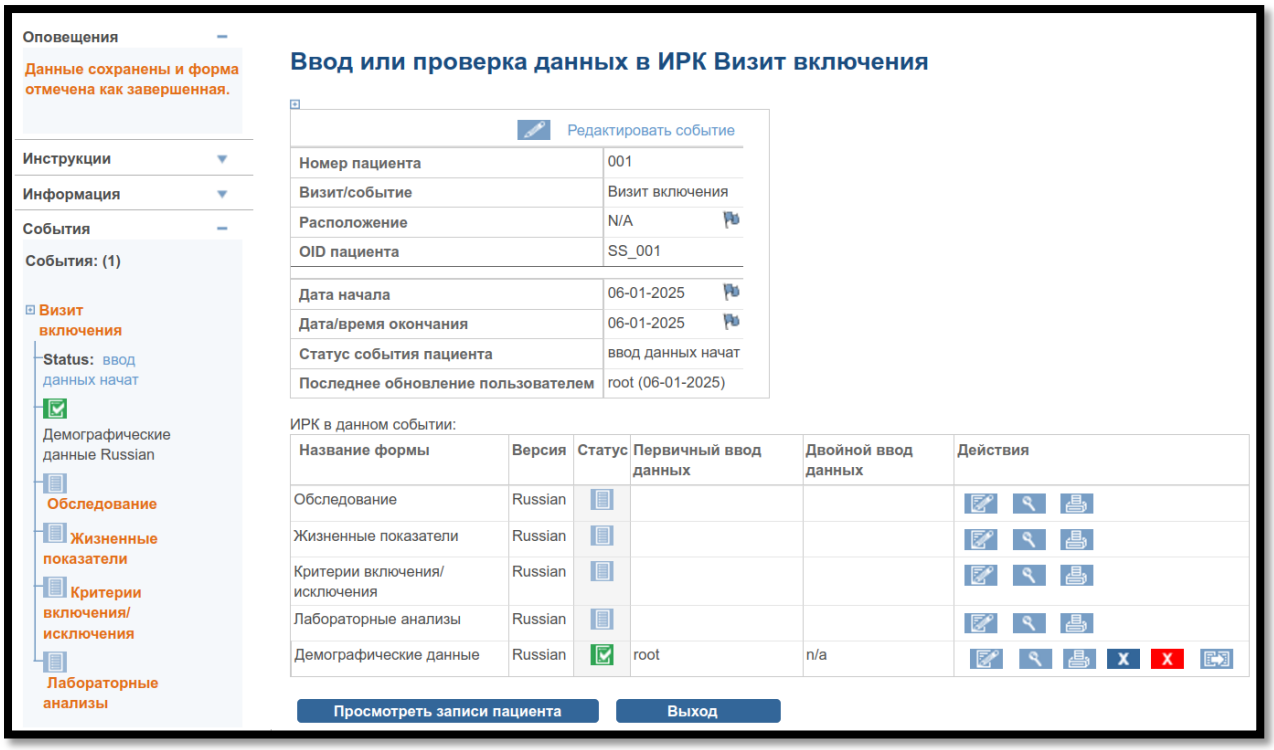


Рисунок 8.236. Страница события “Визит включения”, статус “Ввод данных завершен” для ИРК “Демографические данные”

Далее следует вернуться к записи пациента с помощью кнопки “Просмотреть записи пациента” и создать новое событие “Лабораторное исследование”, нажав кнопку “Запланировать новый визит/событие”.

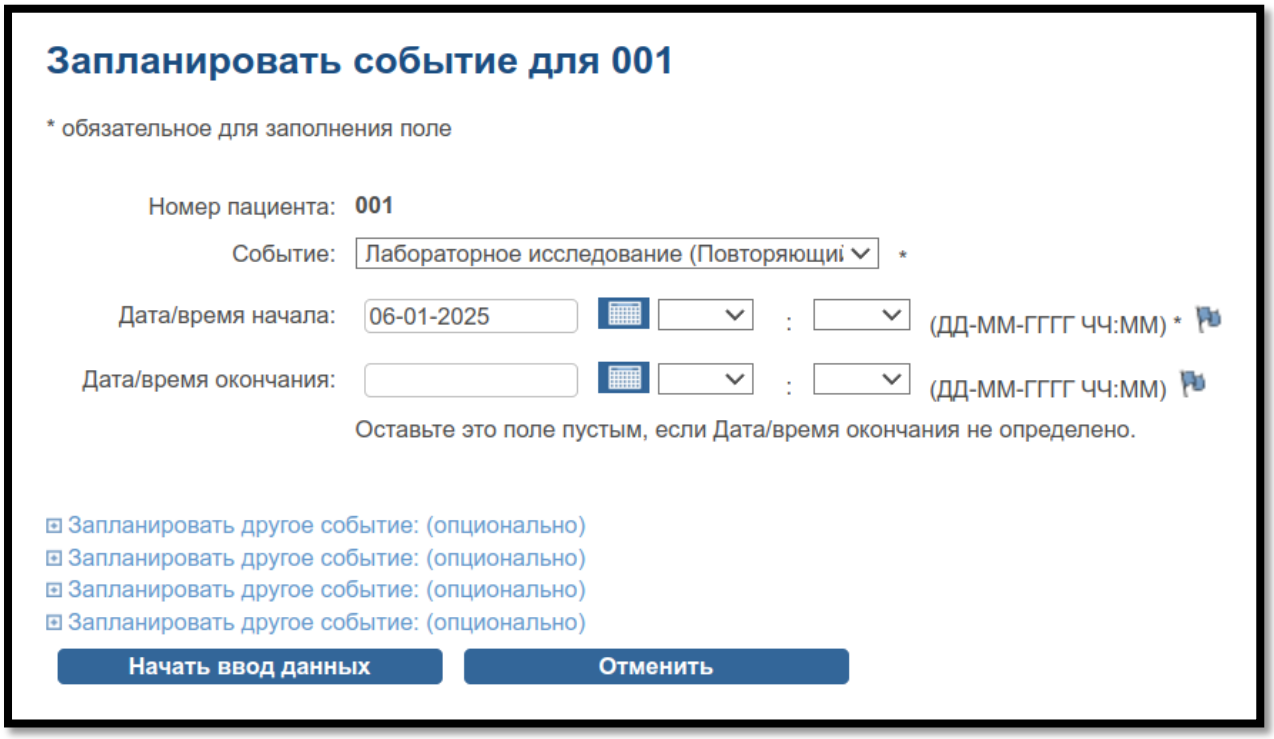


Рисунок 8.237. Страница создания нового события события “Лабораторное исследование”

На данной странице необходимо выбрать тип события и дату его начала. Также раскрывая блоки ниже на странице можно создать до пяти событий одновременно. Нажатие кнопки “Начать ввод данных” откроет созданное событие. Далее в качестве примера будет заполнена одна ИРК. Для этого необходимо вернуться к списку пациентов.

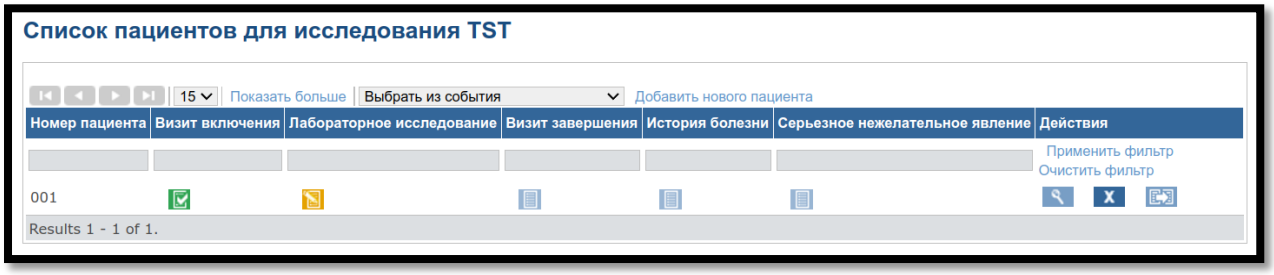


Рисунок 8.238. Страница со списком пациентов

В таблице появилась строка с созданным пациентом, в ячейках указываются статусы событий: “Визит включения” - завершен, “Лабораторное исследование” - ввод данных начат, остальные события - не запланировано. События можно создавать и открывать также с помощью данной таблицы - для этого необходимо кликнуть на иконку со статусом. Например, можно создать еще одно событие “Лабораторное исследование” кликнув на иконку и выбрав пункт “Добавить другое событие”.

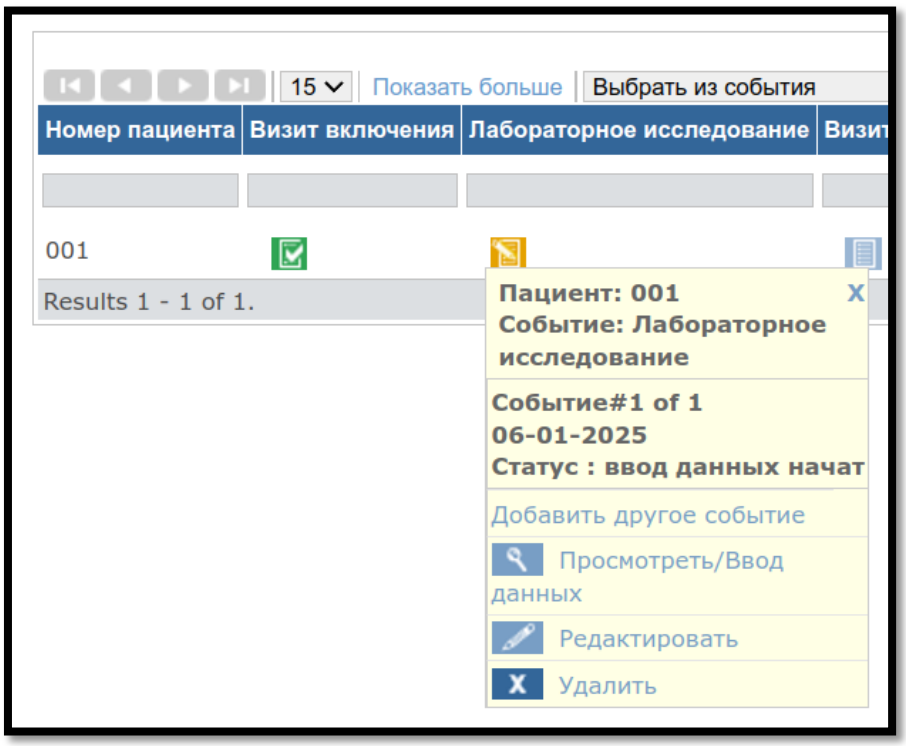


Рисунок 8.239. Добавление дополнительного события “Лабораторное исследование”

В списке пациентов рядом со статусом события появится текст “x2”, показывающий, что у создано два события типа “Лабораторное исследование” (**рисунок 8.240**). Теперь при клике можно выбирать, в какое событие нужно войти, чтобы ввести или удалить данные.

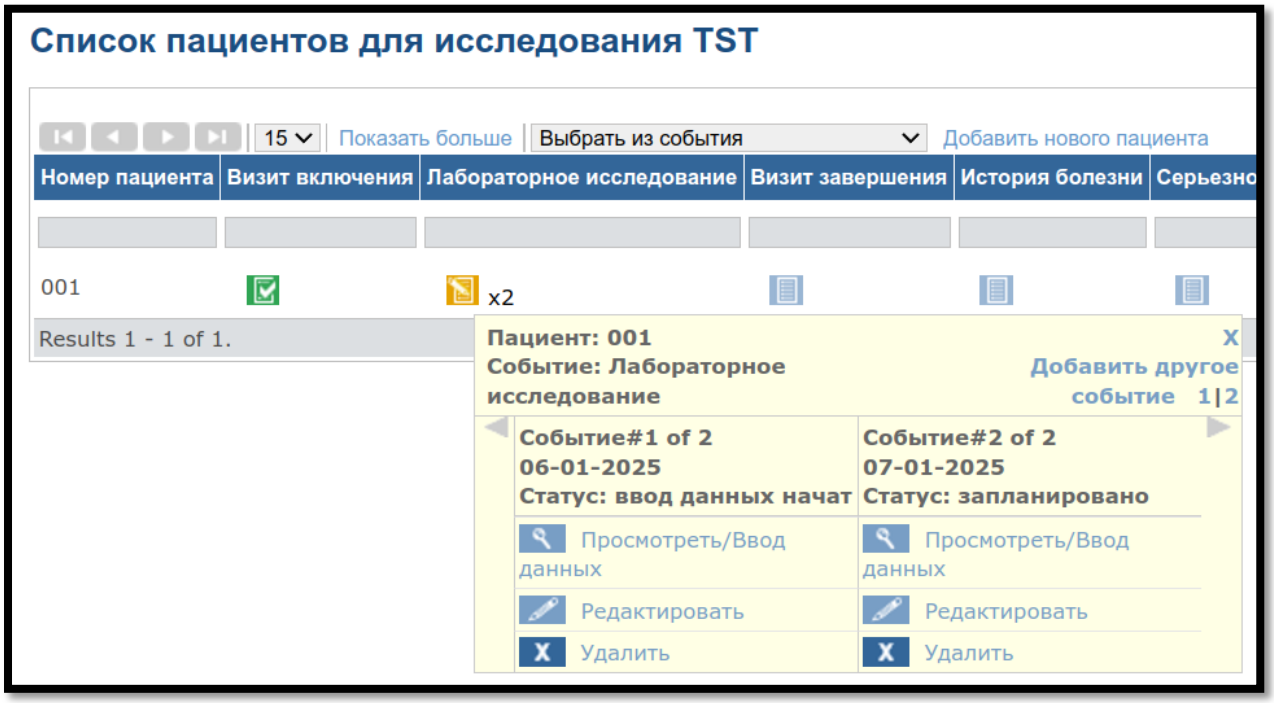


Рисунок 8.240. Два события “Лабораторное исследование”

Следует отметить, что при каждом изменении данных все изменения записываются в “Журнал аудита”. Таким образом, можно проследить всю историю изменений. Поэтому при удалении событий или пациентов данные на самом деле не удаляются, а им только присваивается статус удаленных, но они остаются видны в списке пациентов и при необходимости их можно восстановить.

Экспорт данных из LibreClinica

После того, как все процедуры ввода, верификации и валидации были завершены, а все ИРК подписаны главным исследователем можно приступать к процедуре экспорта данных для их анализа и обработки.

Для этого необходимо выбрать в меню “Действия” (**рисунок 8.241**), раздел “Извлечь данные” пункт “Создать набор данных”.

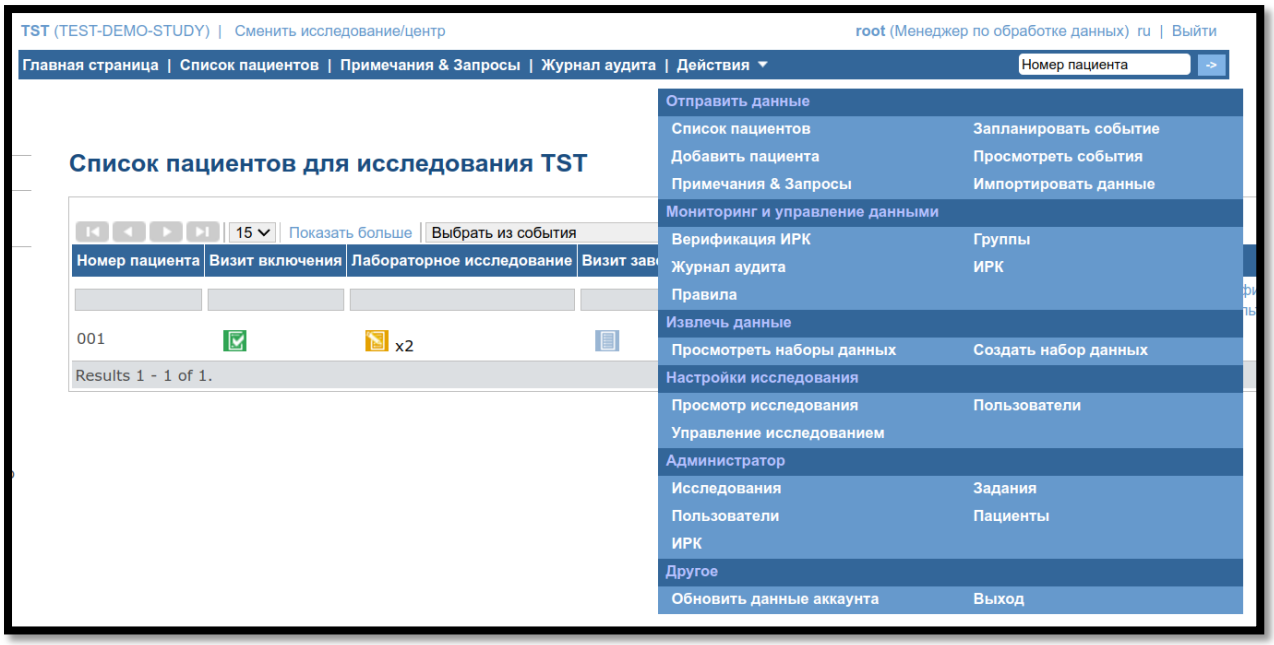


Рисунок 8.241. Меню “Действия”

При выборе пункта "Создать набор данных" запускается мастер (**рисунок 8.242**), в котором последовательно необходимо отметить, какие именно поля из каких ИС следует включить в результирующий набор, а также какие атрибуты событий и пациентов. Для упрощения процесса выбора имеется кнопка "Выбрать все переменные".

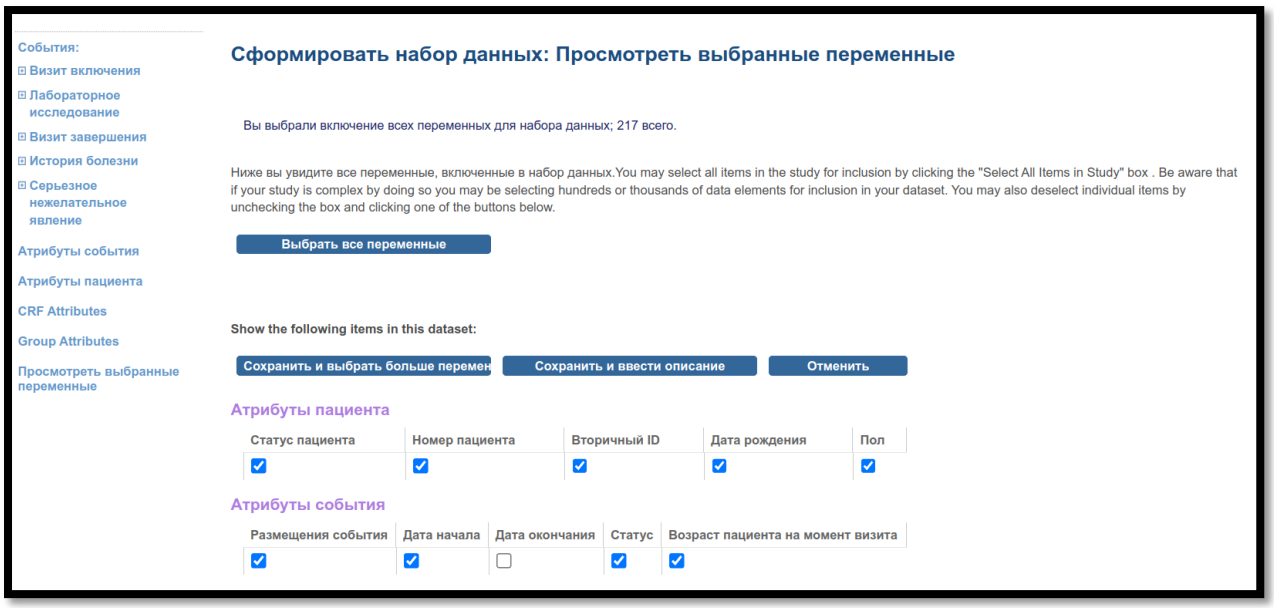


Рисунок 8.242. Мастер экспорта данных

После завершения выбора необходимых параметров нужно сохранить набор данных и задать временной период, за который требуется выгрузить данные. Необходимо

учитывать именно даты начала событий, как они были записаны в системе, а не фактические даты их внесения. Для полной выгрузки рекомендуется задавать максимально возможный диапазон.

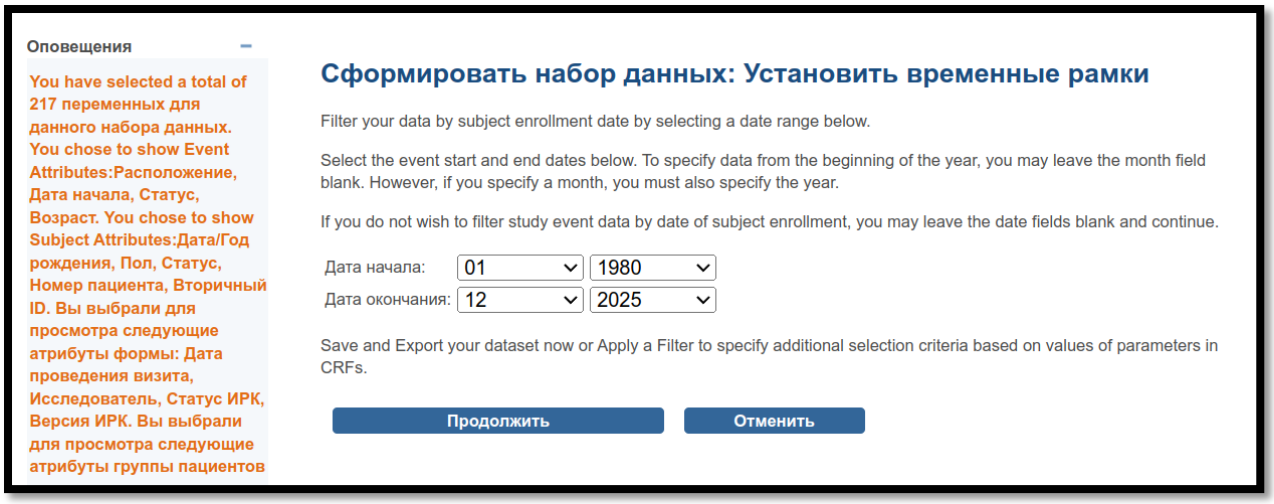


Рисунок 8.243. Выбор диапазона дат для экспорта данных

На финальном этапе необходимо ввести название набора данных и выбрать из каких ИРК требуется выгружать данные (**рисунок 8.244**).

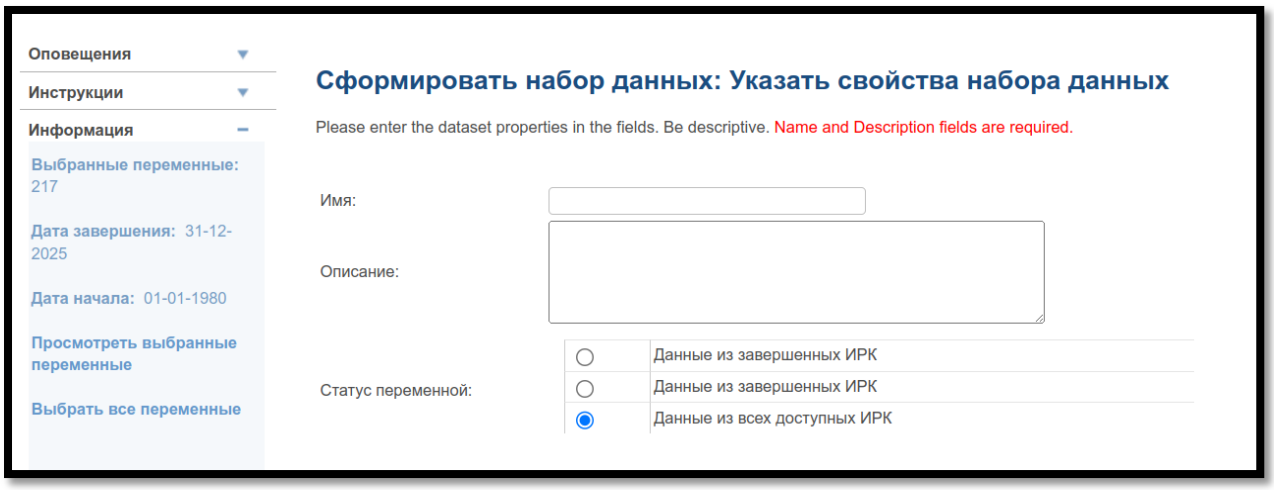


Рисунок 8.244. Указание название набора данных и выбор ИРК для экспорта

После сохранения можно считать, что структура набора данных создана, а именно определено, какие данные и за какой период необходимо извлечь. Сам по себе такой набор данных не содержит моментальный снимок данных, а лишь описывает, как этот снимок будет получен. Преимущество данного подхода заключается в том, что достаточно один раз выбрать переменные, необходимые для извлечения, а затем периодически выполнять извлечение по мере заполнения базы данных в ходе исследования. Таким образом, будет постоянно получаться актуальный снимок данных без необходимости повторного создания набора данных. Все созданные снимки

данных сохраняются в системе, и их можно повторно загрузить при необходимости (**рисунок 8.245**).

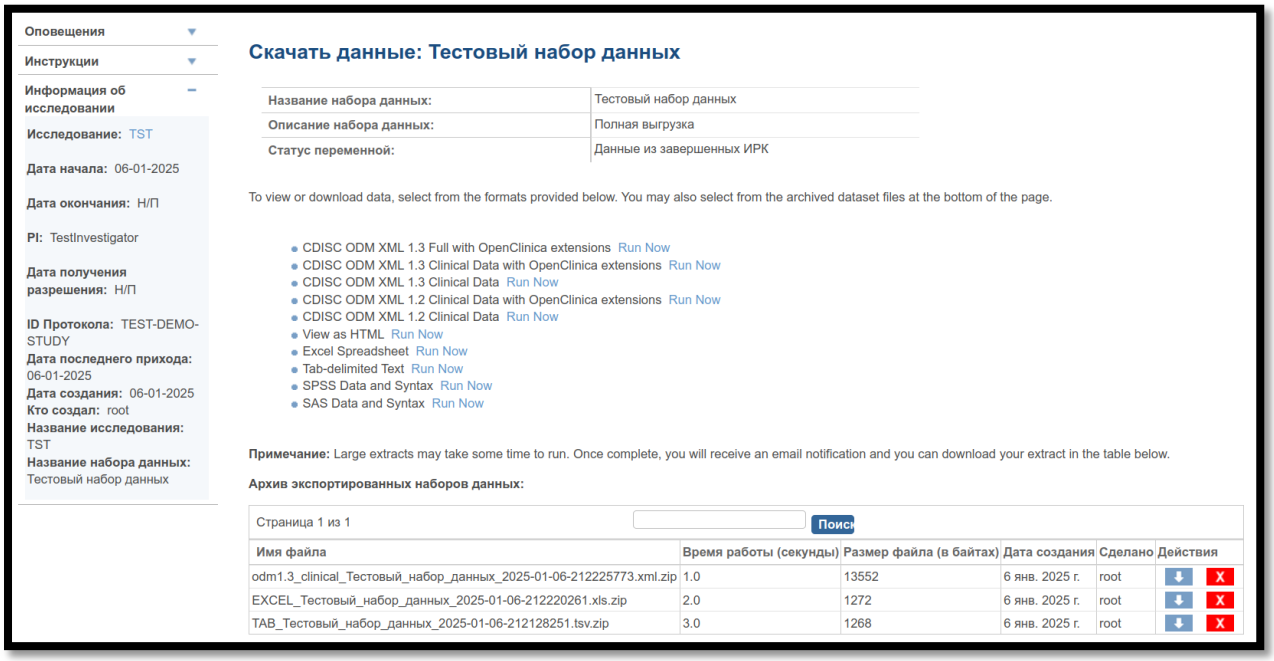


Рисунок 8.245. Список снимков данных для экспорта

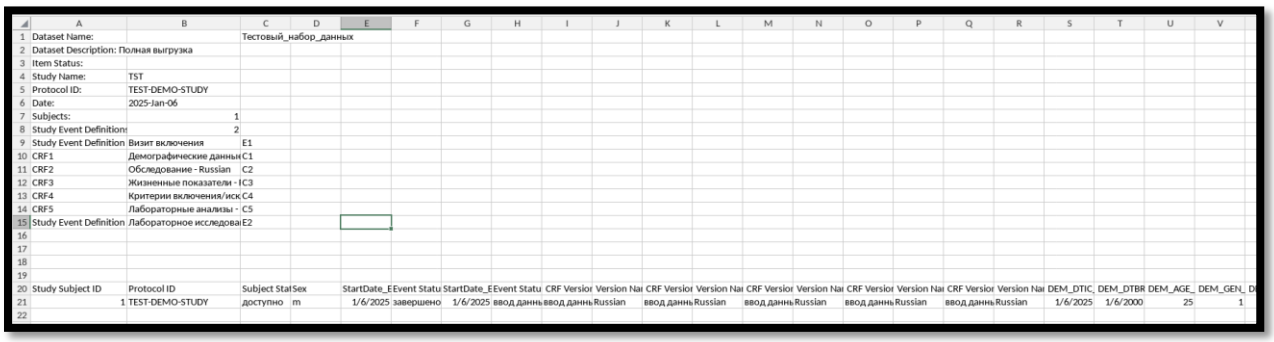
Список доступных форматов для скачивания указан в верхней части **рисунка 8.245**. Наиболее компактными форматами являются форматы Excel Spreadsheet, Tab-delimited Text, SPSS Data and Syntax, SAS Data and Syntax, так как все они содержат примерно одинаковую таблицу, где в каждой строке записана информация по одному пациенту, а коды переменных расположены в столбцах (**рисунки 8.246-8.247**).

```
Dataset Name:       Тестовый_набор_данных
Dataset Description: Полная выгрузка
Item Status:
Study Name:     TST
Protocol ID:    TEST-DEMO-STUDY
Date:   2025-Jan-06
Subjects:   1
Study Event Definitions 2
Study Event Definition 1    Визит включения E1
CRF1    Демографические данные - Russian    C1
CRF2    Обследование - Russian  C2
CRF3    Жизненные показатели - Russian  C3
CRF4    Критерии включения/исключения - Russian C4
CRF5    Лабораторные анализы - Russian  C5
Study Event Definition 2    Лабораторное исследование   E2



Study Subject ID    Protocol ID Subject Status  Sex StartDate_E1    Event Status_E1 StartDate_E2_1  Event Status_E2_1
CRF Version Status_E1_C1     Version Name_E1_C1  CRF Version Status_E1_C2    Version Name_E1_C2  CRF Version
Status_E1_C3     Version Name_E1_C3  CRF Version Status_E1_C4    Version Name_E1_C4  CRF Version Status_E1_C5
Version Name_E1_C5   DEM_DTIC_E1_C1      DEM_DTBR_E1_C1      DEM_AGE_E1_C1   DEM_GEN_E1_C1   DEM_ETH_E1_C1
DEM_RACE_E1_C1   DEM_MAR_E1_C1   DEM_EDU_E1_C1   APPEARANCE_E1_C2    SKIN_E1_C2      HEENT_E1_C2     THYROID_E1_C2
CHEST_E1_C2      LUNGS_E1_C2     BREAST_E1_C2    HEART_E1_C2     ABDOMEN_E1_C2   MUSCULOSKELETAL_E1_C2
GENITALIA_E1_C2      PELVIS_E1_C2    RECTAL_E1_C2    PROSTATE_E1_C2      VASCULAR_E1_C2      NEUROLOGICAL_E1_C2
LYMPHNODES_E1_C2     VS_TEMP_E1_C3   VS_TUNIT_E1_C3      VS_HGHT_E1_C3   VS_WGHT_E1_C3   VS_UNITS_E1_C3
VS_BMI_E1_C3     VS_BPHR_E1_C3   IN_NINCDS_E1_C4     IN_MMSE_E1_C4   IN_NMDA_E1_C4   IN_HOME_E1_C4   IN_CARE_E1_C4
IN_ENGSP_E1_C4   IN_STBL_E1_C4   IN_DOSE_E1_C4   IN_HEPIN_E1_C4      IN_ORAL_E1_C4   IN_IC_E1_C4     IN_DT_IC_E1_C4
EX_OGTT_E1_C4    EX_HEP_E1_C4    EX_CARD_E1_C4   EX_RECNT_E1_C4      EX_STRKE_E1_C4      EX_SEIZ_E1_C4   EX_MENTL_E1_C4
EX_ORAL_E1_C4    EX_INS_E1_C4    EX_DRUG_E1_C4   INEX_ELIG_E1_C4     SITE_AB_E1_C5   SITE_DEC_E1_C5      HEM_DT_E1_C5
HEM_HEM_E1_C5    HEM_RBC_E1_C5   HEM_WBC_E1_C5   HEM_RHE_E1_C5   CH_DT_E1_C5     CH_POT_E1_C5    CH_CAL_E1_C5
CH_ALB_E1_C5     UR_DT_E1_C5     UR_PRO_E1_C5    UR_LEUK_E1_C5   UR_NITR_E1_C5   CRF Version Status_E2_1_C6  Version
Name_E2_1_C6     VS_TEMP_E2_1_C6     VS_TUNIT_E2_1_C6    VS_HGHT_E2_1_C6     VS_WGHT_E2_1_C6     VS_UNITS_E2_1_C6
VS_BMI_E2_1_C6   VS_BPHR_E2_1_C6
001 TEST-DEMO-STUDY доступно    m   2025-01-06  завершено   2025-01-06  ввод данных начат   ввод данных завершен
Russian ввод данных завершен    Russian ввод данных завершен    Russian ввод данных завершен    Russian ввод данных
завершен    Russian 2025-01-06  2000-01-06  25  1   1   6   1   5   1   1   1   1   1   1   1   1   1   1   1   1   1
1   1   1   1   36.6    1   180 86  2   26.5    2   1   1   1   1   1   1   1   1   1   1   1   2025-01-08  2   2   2
2   2   2   2   2   2   2   1   101 SiteA   2025-01-06  12  12  12  12  2025-01-06  23  34  23  2025-01-06  0   1   1
ввод данных завершен    Russian 36.6    1   180 86  2   26.5    2
```

Рисунок 8.246. Выгрузка в формате Tab-delimited Text



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Dataset Name:		Тестовый_набор_данных																			
2	Dataset Description: Полная выгрузка																					
3	Item Status:																					
4	Study Name:	TST																				
5	Protocol ID:	TEST-DEMO-STUDY																				
6	Date:	2025-Jan-06																				
7	Subjects:	1																				
8	Study Event Definition	2																				
9	Study Event Definition	Визит включения	E1																			
10	CRF1	Демографические данны	C1																			
11	CRF2	Обследование - Russian	C2																			
12	CRF3	Жизненные показатели - I	C3																			
13	CRF4	Критерии включения/иск	C4																			
14	CRF5	Лабораторные анализы -	C5																			
15	Study Event Definition	Лабораторное исследова	E2																			
16																						
17																						
18																						
19																						
20	Study Subject ID	Protocol ID	Subject Stat	Sex	StartDate_E	Event Statu	StartDate_E	Event Statu	CRF Versio	Version Na	CRF Versio	Version Na	CRF Versio	Version Na	CRF Versio	Version Na	CRF Versio	Version Na	DEM_DTIC_	DEM_DTBR	DEM_AGE_	DEM_GEN_ D
21	1	TEST-DEMO-STUDY	доступно	m	1/6/2025	завершено	1/6/2025	ввод данны	ввод данны	Russian	ввод данны	Russian	ввод данны	Russian	ввод данны	Russian	ввод данны	Russian	1/6/2025	1/6/2000	25	1
22																						

Рисунок 8.247. Выгрузка в формате Excel Spreadsheet

Представленные форматы достаточно удобны для обработки, однако не включают описания, что именно обозначают данные переменные, поэтому для полноценной работы требуются либо исходные файлы ИРК, либо дополнительная информация из LibreClinica. Поэтому для создания максимально автономной и самодостаточной выгрузки наилучшим образом подходит формат CDISC ODM XML, так как он кроме непосредственно данных включает и определения всех полей, событий и параметров исследования.