**Санкт‑Петербургское государственное унитарное предприятие**

**«Санкт‑Петербургский информационно‑аналитический центр»**

**(СПб ГУП «СПб ИАЦ»)**

ПРОГРАММА ДЛЯ ЭВМ

**ЭЛЕКТРОННЫЙ ЛЬГОТНЫЙ РЕЦЕПТ**

РУКОВОДСТВО АДМИНИСТРАТОРА

На 10 листах

Содержание

1 Общие сведения о программе 3

1.1 Назначение программы 3

1.2 Функции программы 3

1.3 Минимальный состав технических средств 3

1.4 Минимальный состав программных средств 4

2 Структура программы 5

2.1 Сведения о структуре программы 5

3 Настройка программы 6

4 Проверка программы 7

4.1 Проверка технического состояния программы 7

4.1.1 Сервер базы данных 7

4.1.2 Сервер приложений 7

4.1.3 Прокси‑сервер 7

4.2 Проверка установленного ПО 8

5 Дополнительные возможности 9

5.1 Процедура запуска/останова программы 9

5.1.1 Процедура запуска сервера базы данных 9

5.1.2 Процедура останова сервера базы данных 9

5.1.3 Процедура запуска сервера приложений 9

5.1.4 Процедура останова сервера приложений 9

5.1.5 Процедура запуска прокси‑сервера 10

5.1.6 Процедура останова прокси‑сервера 10

5.2 Ограничение доступа на время проведения регламентного обслуживания или устранения критических ошибок 10

# Общие сведения о программе

## Назначение программы

Программа предназначена для информационного сопровождения процесса обеспечения граждан льготным лекарствами.

## Функции программы

Основные функции программы:

Регистрация рецепта и печать рецептов на бланках формы N 148-1/У-04(Л) (Приказ Минздрава РФ от 24.11.2021 N 1094Н) и Форма N 1-МИ (Приказ Минздрава РФ от 20.12.2012 N 1181н).

Генерация и вывод статистических отчетов по деятельности МО и региона по зарегистрированным рецептам.

Обмен данными с фармацевтическими организациями (фармацевтические базы и аптечные организации) о выписанных и отпущенных рецептах, об остатках лекарственных препаратов в фармацевтических организациях, движению лекарственных препаратов между фармацевтическими организациями.

Создание, подпись и хранение структурированных электронным медицинских документов (Отпуск по рецепту на лекарственный препарат, изделие медицинского назначения и специализированный продукт лечебного питания (CDA) Редакция 4 - 1.2.643.5.1.13.13.15.2.4 и Льготный рецепт на лекарственный препарат, изделие медицинского назначения и специализированный продукт лечебного питания (CDA) Редакция 4 - 1.2.643.5.1.13.13.15.1.4).

Создание механизмов выписки и отпуска электронных(безбумажных) рецептов.

Сбор данных для формирования потребности в закупке лекарственных препаратов на год, следующий за отчетным.

Обмен с Единой государственной информационной системой здравоохранения данными выписанных и отпущенных рецептах, остатках ЛП в пунктах отпуска.

Управление данными справочников (внесение, изменение, удаление, хранение, распространение) врачей, медицинских организаций, фармацевтических организаций, лекарственных препаратов и агрегация данных справочниками с федеральной НСИ.

## Минимальный состав технических средств

Программа может функционировать на технических средствах, удовлетворяющих требованиям, перечисленным в таблице 1.

Таблица 11 Требования к техническим средствам

| Назначение | Технические характеристики |
| --- | --- |
| 1 | 2 |
| Сервер базы данных | Архитектура: x86‑64;ЦПУ: не менее 2 ядер, частота не менее 2,4 ГГц;ОЗУ: не менее 4 Гбайт;дисковое пространство: не менее 16 Гбайт для установки базового ПО, дополнительно выделенный том для размещения данных – не менее 50 Гбайт;резервных копий данных – не менее 100Гбайт;сетевой интерфейс: 1 Гбайт/c |
| Сервер приложений | Архитектура: x86‑64;ЦПУ: не менее 2 ядер, частота не менее 2,4 ГГц;ОЗУ: не менее 8 Гбайт;дисковое пространство: не менее 16 Гбайт для установки базового ПО;дополнительно выделенный том для хранения исполняемых файлов – не менее 20 Гбайт;дополнительно выделенный том для размещения данных – не менее 50 Гбайт;сетевой интерфейс: 1 Гбайт/c |
| Прокси‑сервер | Архитектура: x86‑64;ЦПУ: не менее 1 ядра, частота не менее 1,6 ГГц;ОЗУ: не менее 1 Гбайт;дисковое пространство: не менее 16 Гбайт для установки базового ПО;сетевой интерфейс: 1 Гбайт/c |
| АРМ пользователей | Процессор – не менее 2 ядер 1,5 ГГц или совершеннее;оперативная память не менее 4096 Мбайт;видеокарта и монитор с разрешающей способностью не менее 1280\*1024;жесткий диск объемом памяти не менее 120 Гбайт;сетевая карта Ethernet 10/100 Мбит/с;источник бесперебойного питания |

## Минимальный состав программных средств

Требования к программным средствам приведены в таблице 2.

Таблица 22 Требования к программным средствам

| Назначение | Общее ПО |
| --- | --- |
| 1 | 2 |
| Сервер базы данных | ОС: Debian 11.и более новые;СУБД: PostgreSQL выпуск 11 и более новые |
| Сервер приложений | ОС: Debian 11.и более новые;dotnet-sdk , Mono 6.12 |
| Прокси‑сервер | ОС: Debian 11. и более новые;Nginx 1.13 и более новые |
| АРМ пользователей | Microsoft Windows 7/8/8.1/10 / Linux OS с графическим пользовательским интерфейсом веб‑браузер: Mozilla Firefox 54 и более современные версии (Google Chrome, Microsoft Edge, Яндекс Браузер) |

# Структура программы

## Сведения о структуре программы

Программа построена на основе трехзвенной архитектуры, с выделенными уровнями сервера базы данных, сервера приложений, и рабочих мест пользователей, организованных с использованием «тонкого клиента».

Описание структуры и алгоритма программы приведены в документе «Программа для ЭВМ «Электронный льготный рецепт». Описание программы».

# Настройка программы

Для функционирования программы требуется установить на серверы программные средства, приведенные в п. 1.4.

Для упрощения процедуры установки, перед ее началом требуется заполнить таблицу данными, которые пригодятся в процессе установки. Шаблон для заполнения приведен в таблице 3.

Таблица 33 Шаблон с данными, необходимыми в процессе установки

| Роль | DNS‑имя сервера | IP‑адрес основного интерфейса |
| --- | --- | --- |
| 1 | 2 | 3 |
| Прокси‑сервер | proxy.dns.suffix | w.x.y.z |
| Сервер приложений | prefix‑app.dns.suffix | w.x.y.z |
| Сервер базы данных | db.dns.suffix | w.x.y.z |

Установка программы включает следующие шаги:

1. На всех серверах для выполнения и унификации настройки создать локального пользователя sysop с домашним каталогом и возможностью повышения привилегий в системе с помощью утилиты sudo до уровня пользователя root.
2. На сервере приложений переключиться на работу под пользователем sysop.
3. Сгенерировать ключи для асимметричного шифрования подключений с помощью протокола SSH:

ssh-keygen

1. Скопировать публичный ключ пользователя sysop на сервер баз данных, заменив в примере ниже DNS‑имя сервера БД из таблицы:

ssh-copy-id sysop@db.dns.suffix

1. Установить на сервер приложений программу автоконфигурации ansible версии не ниже 2.8 (сайт разработчика: http://ansible.com). Этот шаг требует повышения привилегий до уровня root.
2. На сервере ci запустить задачу prod\_prepare\_servers с ключами:

dbservers\_pg (db.dns.suffix);

appservers (prefix‑app.dns.suffix);

proxyservers (proxy.dns.suffix).

1. На сервере непрерывной интеграции запустить задачи в процессе выполнения указав версии устанавливаемого СПО:

prod-library-release

prod-event-release

prod-stage-release-pipeline

1. Проверить вывод запущенных задач на наличие ошибок, при необходимости внести изменения и перезапустить.

# Проверка программы

После выполнения установки и настройки программы проверка работоспособности осуществляется в два этапа.

## Проверка технического состояния программы

Выполняется непосредственно на серверах. Требует повышения привилегий до уровня пользователя root.

### Сервер базы данных

Проверка программы на сервере баз данных включает следующие шаги:

1. Открыть интерактивный сеанс работы с сервером (консоль или ssh‑сессия).
2. Выполнить проверку наличия рабочих процессов и открытых сетевых сокетов:

# Выводит перечень процессов БД postgres

# Ожидаемое значение в выводе: строка с идентификатором процесса и временем его активности

ps -aef | grep postmaster

# пример выполнения:

# postgres 25818 1 0 2019 ? 05:57:45 /usr/pgsql-11/bin/postmaster -D /data/pgsql/data

# выводит список прослушиваемых сетевых сокетов

# Ожидаемое значение: State=LISTEN, Local Address:Port=:::5432

ss -tln|egrep -i '5432|state'

# пример выполнения:

State Recv-Q Send-Q Local Address:Port Peer Address:Port

LISTEN 0 128 \*:5432 \*:\*

LISTEN 0 128 [::]:5432 [::]:\*

### Сервер приложений

Проверка программы на сервере приложений включает следующие шаги:

1. Открыть интерактивный сеанс работы с сервером (консоль или ssh‑сессия).
2. Выполнить проверку наличия рабочих процессов и открытых сетевых сокетов:

# Выводит перечень java-процессов

# Ожидаемое значение в выводе строка с идентификатором процесса и временем его активности

ps -aef | grep java

# пример выполнения:

1401 25088 1 2 Sep25 ? 1-00:53:14 /opt/java/bin/java -Dnop -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Dfile.encoding=UTF8 -Djava.net.preferIPv4Stack=true -Dorg.apache.catalina.loader.WebappClassLoader.ENABLE\_CLEAR\_REFERENCES

# выводит список прослушиваемых сетевых сокетов

# Ожидаемое значение: State=LISTEN, Local Address:Port=:::8080

ss -tln|egrep -i '8080|state'

State Recv-Q Send-Q Local Address:Port Peer Address:Port

LISTEN 0 100 :::8080 :::\*

### Прокси‑сервер

Проверка программы на прокси‑сервере включает следующие шаги:

1. Открыть интерактивный сеанс работы с сервером (консоль или ssh‑сессия).
2. Выполнить проверку наличия рабочих процессов и открытых сетевых сокетов:

# Выводит перечень процессов прокси-сервера

# Ожидаемое значение в выводе строка с идентификатором процесса и временем его активности

ps -aef | grep nginx

# пример выполнения:

root 2699 1 0 Jul21 ? 00:00:00 nginx: master process /usr/sbin/nginx -c /etc/nginx/nginx.conf

# выводит список прослушиваемых сетевых сокетов

# Ожидаемое значение: State=LISTEN, Local Address:Port=:::443

ss -tln|egrep -i '80|443|state'

# пример выполнения:

State Recv-Q Send-Q Local Address:Port Peer Address:Port

LISTEN 0 128 :::80 :::\*

LISTEN 0 128 :::443 :::\*

## Проверка установленного ПО

Выполнить проверку веб‑интерфейса программы, для чего с рабочего места открыть в веб‑браузере адрес настроенного сервера приложений вида <имя целевого сервера>/armcto.

В веб‑интерфейсе проверить основной функционал, описанный в документе «Программа для ЭВМ «Электронный льготный рецепт». Руководство пользователя».

# Дополнительные возможности

Для выполнения операций необходим интерактивный сеанс работы с сервером (консоль или ssh‑сессия), в котором команды выполняются с привилегиями root, если не сказано иное.

## Процедура запуска/останова программы

### Процедура запуска сервера базы данных

Процедура запуска сервера базы данных выполняется с привилегиями root, если не указано иное и включает следующее:

1. Включить сервер (виртуальную машину).
2. Проверить отсутствие ошибок (устранить, если обнаружены) запуска ОС и системных сервисов с помощью журналов: /var/log/boot.log, var/log/dmesg, /var/log/messages.
3. Проверить открытые сетевые порты баз данных:

# ss -tln|egrep -i '5432|state'

1. Проверить журнал предупреждений базы данных на предмет наличия ошибок:

# less /data/pgsql/data/log/postgresql-\*.log

1. Проверить процессы требуемых сервисов базы данных:

$ systemctl status postgresql-11

Вывод должен содержать имя БД и статус «READY»

### Процедура останова сервера базы данных

Процедура останова сервера баз данных выполняется с привилегиями root, если не указано иное и включает следующее:

1. Проверить статус БД (выполняется с привилегиями root):

$ systemctl status postgresql-11

Вывод должен содержать имя БД и статус «READY»

1. Выполнить команду:

$ systemctl stop postgresql-11

1. Проверить список процессов пользователя postgres:

# pgrep -u postgres

В случае обнаружения процессов экземпляра БД выяснить причину их наличия.

1. Выполнить команду останова ОС:

# init 0

### Процедура запуска сервера приложений

Процедура запуска сервера приложений выполняется с привилегиями root, если не указано иное и включает следующее:

1. Включить сервер (виртуальную машину).
2. Проверить отсутствие ошибок (устранить, если обнаружены) запуска ОС и системных сервисов с помощью журналов: /var/log/messages, var/log/dmesg, var/log/syslog.
3. Запустить работу:

dotnet <название исполняемого файла> --urls=http://\*:808

1. Проверить наличие процессов требуемых сервисов:

# ps -ef | grep Lrec

1. Проверить доступность приложения по сети:

http:// <имя целевого сервера>/ Lrec/

1. Проверить журнал приложения на предмет наличия ошибок:

# less /data/logs/liferay.log

### Процедура останова сервера приложений

Процедура останова сервера приложений выполняется с привилегиями root, если не указано иное и включает следующее:

1. Остановить работу контейнера:

# nginx stop

1. Проверить наличие процессов останавливаемых сервисов:

# cto -stop

В случае обнаружения процессов выяснить причину их наличия.

1. Для полного останова сервера выполнить команду останова ОС:

# init 0

### Процедура запуска прокси‑сервера

Процедура запуска прокси‑сервера выполняется с привилегиями root, если не указано иное и включает следующее:

1. Включить сервер (виртуальную машину).
2. Проверить отсутствие ошибок (устранить, если обнаружены) запуска ОС и системных сервисов с помощью журналов: /var/log/boot.log, var/log/dmesg, /var/log/messages, /var/log/syslog.
3. Проверить процессы требуемых сервисов:

# systemctl status nginx

1. Проверить открытые сетевые порты:

# ss -tln | egrep '80|443'

1. Проверить журнал на предмет наличия ошибок:

# less /var/log/nginx/error.log

Проверить доступность веб‑страницы http://<hostname>.

### Процедура останова прокси‑сервера

Процедура останова прокси-сервера выполняется с привилегиями root, если не указано иное и включает следующее:

1. Выполнить команду останова сервиса:

systemctl stop nginx

1. Проверить состояние останавливаемого сервиса:

 systemctl status nginx

В случае обнаружения процессов выяснить причину их наличия.

1. Выполнить команду останова ОС:

# init 0

## Ограничение доступа на время проведения регламентного обслуживания или устранения критических ошибок

Для ограничения доступа необходимо запустить задачу на сервере непрерывной интеграции с опцией «true»:

install\_stub

По окончании выполнения требуемых работ необходимо восстановить доступ запустив задачу на сервере непрерывной интеграции с опцией «false»:

install\_stub