

# Техническое задание: «Старт с любого экрана» (добавка к проекту)

Проект: AOzonJobTracker (Ozon Job Watcher). Проект уже существует - добавляем только эту фичу и связанные страховки.

Дата: 2026-02-07

Суть фичи: в начале каждого цикла проверки бот должен уметь начать маршрут из любой точки внутри Ozon Job. Он распознаёт текущий экран (по якорям), выбирает самый короткий и безопасный путь к цели или делает восстановление (recovery). Дальше используется существующая проверка слотов для «Петровское» -> «Записаться» -> «Производство непрофиль».

## Маршрут (как у пользователя сейчас):

- В нижнем меню нажать «Склады».
- На экране «Выберите склад» в поиске ввести «Петровское».
- Открыть карточку склада «Петровское».
- Нажать кнопку «Записаться».
- В списке работ найти «Производство непрофиль» и проверить, есть ли даты/нет ли «НЕТ МЕСТ».

## 0. Границы и принципы безопасности

- Фи́ча не добавляет автозапись на слот. Только навигация к экрану проверки и анализ доступности.
- Никаких обходов логина/2FA/капч. Если приложение запросило ввод - бот останавливается и сообщает пользователю.
- Никаких хаотичных кликов: действие выполняется только если экран распознан с достаточной уверенностью или внутри управляемого recovery.
- Если проект когда-либо будет публиковаться в Google Play: для Accessibility есть отдельные требования и ограничения.

## 1. Что должно измениться в проекте (выходные артефакты)

- Добавить ScreenDetector (распознавание экранов по якорям).
- Добавить Router (решает, в какой state перейти и какое действие сделать сейчас).
- Добавить RecoveryManager (Back -> Home+relaunch, лимиты, Safe Mode).
- Врезать новый входной шаг BOOTSTRAP\_DETECT\_AND\_ROUTE в существующий state machine.
- Доработать логи: распознанный экран, confidence, решение, recovery шаги.

## 2. Требования к функционалу

ID	Требование	Проверка
F1	Старт из любого экрана внутри ru.ozon.hire	В начале каждого цикла: detect -> route -> действие или recovery.
F2	Короткий путь, если уже близко к цели	Если уже открыт «Петровское: список работ» - без лишних шагов сразу к «Производство непрофиль».
F3	Безопасное восстановление	Если экран неизвестен/низкая уверенность - управляемый recovery с лимитами.
F4	Нижняя навигация как «якорь безопасности»	Если видна панель (Записи/Выплаты/Склады/...) - первым действием безопасно перейти в «Склады».
F5	Wait-until вместо слепых задержек	На 3 ключевых переходах: ждать якорь с таймаутом; при провале - recovery.
F6	Диагностика	Лог на цикл: screen_id/confidence/решение/шаги/результат.

## 3. ScreenDetector

ScreenDetector берёт текущий Accessibility root (или список окон, если включено retrieveInteractiveWindows) и возвращает: ScreenId, confidence (0..1), matched\_anchors.

### 3.1 Набор экранов (MVP + расширения)

MVP - 4 экрана, чтобы закрыть то, что ты описал и прислал скриншотами. Расширения можно добавлять позже (например, экран выплат/поездов).

ScreenId	Экран	Как распознаём (якоря)	Действие
RECORDS_MAIN	Записи (главный)	Якоря: «Записи» (заголовок) + любой из: «Календарь», «Горящие слоты»; + наличие нижней панели.	Тар «Склады»
WAREHOUSES_LIST	Выберите склад	Якоря: «Выберите склад» + строка поиска + список карточек складов.	Поиск «Петровское»
WAREHOUSE_CARD_PETROVSKOE	Карточка склада	Якоря: заголовок «Петровское» + кнопка «Записаться» (крупная).	Тар «Записаться»
WORKS_LIST_PETROVSKOE	Список работ	Якоря: заголовок «Петровское» + список работ; часто есть секция «НЕТ МЕСТ».	Искать «Производство непрофиль»
UNKNOWN	Не распознано	Якорей недостаточно или root пустой.	Recovery

### 3.2 Правила распознавания (важно для стабильности)

- Не использовать динамические значения (числа, суммы, рейтинги) как якоря.
- Использовать комбинации: заголовок + уникальная кнопка/поле. Один текст почти всегда недостаточен.

- Ввести порог уверенности: например, `confidence >= 0.7` - «уверенный» экран, иначе UNKNOWN.
- Если совпали якоря двух экранов - выбирать более специфичный (например, наличие «Записаться» сильнее, чем просто «Петровское»).
- Добавить «негативные якоря» при необходимости: например, если видим «Выплаты»/«Поездки» - это не «Склады».

### 3.3 Нюансы Accessibility, которые нужно заложить

- `root` может быть `null` - это нормальная ситуация. В этом случае: без кликов, краткое ожидание и повтор детекта по событию окна.
- Сервис должен иметь доступ к содержимому окна (`canRetrieveWindowContent`).
- Для диалогов/оверлеев может понадобиться `FLAG_RETRIEVE_INTERACTIVE_WINDOWS` и чтение через `getWindows()`.

## 4. Router (план действий из любой точки)

Router получает (`screen_id`, `confidence`, `anchors`) и выбирает следующее действие и целевой state существующего автомата.

### 4.1 Таблица маршрутизации (MVP)

Текущий экран	Действие	Ожидание якоря
RECORDS_MAIN	Нажать нижнюю вкладку «Склады»	Ждать «Выберите склад» / появления поиска
WAREHOUSES_LIST	Продолжить штатный сценарий поиска «Петровское»	Ждать карточку «Петровское»
WAREHOUSE_CARD_PETROVSKOE	Нажать «Записаться»	Ждать список работ «Петровское»
WORKS_LIST_PETROVSKOE	Перейти сразу к поиску «Производство непрофиль» и <code>CHECK_SLOTS</code>	Ждать карточку работы или сделать контролируемый скролл
UNKNOWN	Запустить <code>RecoveryManager</code>	После восстановления - повтор <code>detect+route</code>

### 4.2 Идемпотентность (повтор не должен ломать)

- Если поле поиска уже содержит «Петровское» - не печатать повторно; сравнить строку и только при отличии очистить/ввести заново.
- Если «Петровское» уже открыт и видна кнопка «Записаться» - пропустить «поиск склада».
- Если уже на списке работ - не жать BACK «на всякий случай». Сразу искать/скроллить до «Производство непрофиль».

## 5. RecoveryManager (безопасное восстановление)

Recovery нужен, когда бот стартует на неизвестном экране (или confidence низкий) и нельзя безопасно нажимать элементы.

### 5.1 Алгоритм восстановления (внутри одного цикла)

- Back recovery: нажать BACK до N раз (например 4-6). После каждого BACK: повторить detectScreen(). Если нашли RECORDS\_MAIN или WAREHOUSES\_LIST - остановиться и вернуть управление Router.
- Home+relaunch: если после N BACK всё ещё UNKNOWN - нажать HOME, затем явно открыть Ozon Job (intent).
- Wait-until: ждать базовый якорь (RECORDS\_MAIN или WAREHOUSES\_LIST) с таймаутом. При таймауте - сообщить «нужна ручная помощь», не продолжать клики.

### 5.2 Лимиты и Safe Mode

- Лимит BACK на цикл (N), лимит перезапусков приложения на цикл (M, например 1).
- Если подряд X циклов не удастся распознать экран (например 3) - включать Safe Mode на Y минут: никакой автоматизации, только уведомление/лог.
- Backoff: при повторных провалах увеличивать интервал следующей проверки (например, +60 сек за каждый провал).

### 5.3 Почему обычно ломается (и как лечим)

- root null/пустое дерево - ждать события окна и повторять попытку, не кликать.
- Диалог/оверлей - может быть отдельным окном (требуется retrieveInteractiveWindows).
- Клавиатура/анимации - якоря выбирать по заголовкам/кнопкам, а не по координатам.

## 6. Минимальный Wait-until (3 места)

- После нажатия «Склады»: ждать «Выберите склад»/появление поиска (timeout 6-10 сек).
- После выбора «Петровское»: ждать «Записаться» (timeout 6-10 сек).
- После «Записаться»: ждать список работ/«НЕТ МЕСТ» (timeout 6-10 сек).

Если якорь не появился - recovery вместо продолжения.

## 7. Интеграция в существующий state machine

- Добавить новый входной шаг: BOOTSTRAP\_DETECT\_AND\_ROUTE.
- BOOTSTRAP вызывает ScreenDetector+Router и переводит автомат в существующие состояния:
- - FIND\_WAREHOUSES\_TAB (если RECORDS\_MAIN)
- - FIND\_SEARCH\_FIELD/TYPE\_SEARCH\_QUERY (если WAREHOUSES\_LIST)
- - CLICK\_ENROLL (если карточка «Петровское»)

- - FIND\_JOB\_CARD (если список работ «Петровское»).
- Добавить состояние RECOVERY (или под-состояния) с лимитами.
- Fast Refresh не ломать: recovery не должен бесконечно гонять BACK в режиме fast\_refresh.

## 8. Диагностика и логи (чтобы быстро чинить)

Лог должен отвечать на 3 вопроса: где стартовал? что распознал? почему пошёл так?

- cycle\_id, timestamp
- screen\_id, confidence, matched\_anchors
- router\_decision (куда идём)
- action\_taken (что нажали/где скроллили)
- wait\_anchor\_result (успех/таймаут)
- recovery\_steps (back\_count, relaunch\_count)
- final\_status (success/no\_slots/unknown/manual\_help\_required)

## 9. Тест-кейсы (обязательные)

ID	Сценарий	Ожидаемый результат
T1	Старт на «Записи»	Нажал «Склады» и дошёл до списка работ Петровское
T2	Старт на «Выберите склад»	Сразу поиск Петровское -> карточка склада
T3	Старт на карточке Петровское	Нажал «Записаться» -> список работ
T4	Старт на списке работ Петровское	Сразу ищет/скроллит до «Производство непрофиль»
T5	Старт на неизвестном экране	Back recovery -> Home+relaunch -> база или Safe Mode
T6	root == null 2-3 раза подряд	Нет кликов, ожидание событий, затем recovery/backoff

## 10. Учет ограничений Android/HyperOS

- WorkManager для периодики ограничен минимумом 15 минут - поэтому текущий подход с Foreground Service оправдан.
- Android 12+ ограничивает старт FGS из фона - сервис должен запускаться пользователем и дальше работать как уже запущенный.
- Android 14+ требует корректно указывать foreground service types.
- Xiaomi/HyperOS: Autostart, Battery saver -> No restrictions, закрепление в недавних, разрешение фоновых уведомлений - иначе система может «душить» процесс.

## 11. Acceptance criteria

- Старт с 4 экранов (RECORDS\_MAIN / WAREHOUSES\_LIST / WAREHOUSE\_CARD\_PETROVSKOE / WORKS\_LIST\_PETROVSKOE) работает стабильно.
- UNKNOWN не приводит к случайным кликам: только recovery и, при провале, Safe Mode + уведомление.

- По логам понятно, что произошло на каждом цикле.

## 12. Источники (для разработчика)

Android Developers - Create your own accessibility service

<https://developer.android.com/guide/topics/ui/accessibility/service>

Android source - AccessibilityService.getRootInActiveWindow()

<https://android.googlesource.com/platform/frameworks/base/+96fea02/core/java/android/accessibilityservice/AccessibilityService.java>

Google Codelab - Developing an Accessibility Service

<https://codelabs.developers.google.com/codelabs/developing-android-a11y-service>

Android Developers - WorkManager periodic note (15 min)

<https://developer.android.com/develop/background-work/background-tasks/persistent/getting-started/define-work>

Android Developers - FGS background start restrictions (Android 12+)

<https://developer.android.com/develop/background-work/services/fgs/restrictions-bg-start>

Android Developers - FGS types required (Android 14)

<https://developer.android.com/about/versions/14/changes/fgs-types-required>

Google Play policy - Use of the AccessibilityService API

<https://support.google.com/googleplay/android-developer/answer/10964491>

Don't kill my app - Xiaomi

<https://dontkillmyapp.com/xiaomi>

HyperOS battery saver disable guide (PDF)

<https://www.drivequant.com/hubfs/Tuto%20Smartphones%20-%20EN/Xiaomi/Xiaomi%20-%20HyperOS%20-%20EN.pdf>