# Первичные настройки приложения

При установке приложение должно иметь следующие настройки

1. Идентификатор (номер) каталога по умолчанию
2. Адрес основного веб-сервиса для обмена данными
3. Адрес запасного веб-сервиса, если основной недоступен
4. Частоту проверки обновлений, сообщений и пр. если приложение запущенно и есть доступ к интернету.
5. Адрес для получения поисковых подсказок.

Указанные настройки не доступны пользователю, но доступны для программного изменения, если придет соответствующая команда от сервера.

# Запуск приложения

При первичном запуске приложение должно:

1. Показать «экран загрузки»
2. Выполнить процедуру получения идентификатора сессии
3. Начать процедуру загрузки каталога в фоновом режиме
4. Запросить у пользователя разрешение к геолокации (или вывести уведомление о необходимости разрешить геолокацию, если напрямую из приложения нельзя разрешить)
5. После окончания загрузки показать главный вид приложения.

При последующих запусках приложение должно выполнять следующие действия

1. Выполнить процедуру получения идентификатора сессии
2. Выполнить процедуру запроса обновлений
3. Выполнить запрос загрузки сообщений для пользователя
4. Проверить разрешена ли геолокация, если нет, напомнить пользователю, что надо включить.
5. Отправить данные метрики предыдущего использования программы.
6. Показать главный вид.

# Процедура получения идентификатора сессии

На текущий момент механизм получения идентификатора сессии не проработан. Ниже приводится общее описание.

Процедура получения идентификатора сессии должна выполнять следующие действия:

1. Отправлять запрос на сервис по протоколу https, данные передаются методом POST
2. Дождаться ответа в формате XML который содержит:
   1. Идентификатор сессии
   2. Время действия полученного идентификатора, после которого необходимо повторить процедуру.
   3. Прочую информацию о которой сейчас не догадываемся.

# Процедура загрузки каталога

Для загрузки каталога применяются следующие методы веб-сервиса:

1. GetCatalog() – для загрузки объекта каталога, с настройками, свойствами и пр.
2. GetCatalogNodes() – для получения списка недостающий разделов каталога, если каталог имеет большое кол-во разделов и все они не поместились в один ответ на вызов GetCatalog().
3. GetInterfaceSettings() – для получения списка настроек

Примечание – здесь и далее используются только наименования методов без описания параметров. Точное описание методов можно получить по адресу <http://5.19.235.173:8080/docs/html/>

Процедура загрузки каталога приложением должна выполнять:

1. Отправить запрос GetCatalog
2. После получения, проверить свойство NodeCount полученного объекта Catalog, с кол-вом записей в свойстве Nodes. Если NodeCount > Nodes.Length, произвести догрузку структуры каталога используя метод GetCatalogNodes();
3. Сохранить структуру каталога в локальную БД
4. Загрузить с помощью метода GetInterfaceSettings() – настройки интерфейса, общие для всего интерфейса:
   1. Цвет фона
   2. Цвет шрифта
   3. Изображение фона (если это вообще возможно для мобильных телефонов)
   4. Изображение фона для панели заголовка (если это вообще возможно для мобильных телефонов)
   5. Цвет панели фона заголовка
   6. Изображения иконок для панели навигации
   7. Список будет пополняться.
5. Загрузить и сохранить для дальнейшего использования на телефоне изображения необходимые для отображения каталога: логотип, иконки разделов и пр.

Примечание – на протяжении проекта список необходимого для первичной загрузки будет расширяться, поэтому этот список будет оформлен отдельно. На текущий момент это изображения Catalog.LogoUrl и CatalogNode.Icon

Дальнейшая работа приложения с разделами каталога будет происходить при обращении с локальной копией без обращения к сервису.

# Сбор метрики использования программы

Сбор метрики – сбор информации использования программы, такой как:

1. Время использования программы
2. Карта переходов пользователя в программе
3. Данные геолокации

### Механизм сбора и передачи информации

Сбор осуществляется в ведении текстового лога использования программы, который передается на сервер. После передачи лога и получения ответа, что лог получен, лог очищается и начинает писаться заново.

В логе должны быть отражены все события:

1. Запуск
2. События интерфейса
3. Открытие нового вида
4. Завершение работы программы
5. Переход/Выход в фоновый режим

Передача лога происходит при запуске приложения, если есть доступ к интернету (или сразу после появления доступа к интернету).

Для передачи служит метод веб-сервиса SaveMetrics(). Если в результате вызова приходит ответ true, то можно стирать накопленный лог и начать запись заново, в противном случае, необходимо дописывать в существующий лог.

### Формат лога

1. Формат лога – текст. В кодировке UTF-8
2. Каждое событие - запись из параметров разделенных пробелами
3. Записи разделены знаком «точка с запятой»
4. Все записи в одну строчку (без знаков переноса строки и пр.)

Формат события запуска: yyyy-MM-dd hh:mm:ss Start данные\_геолокации\_разделенные\_пробелом.

1. Формат события завершения работы: yyyy-MM-dd hh:mm:ss End
2. Формат события перехода в фоновый режим: yyyy-MM-dd hh:mm:ss PauseStart
3. Формат события выхода из фонового режима: yyyy-MM-dd hh:mm:ss PauseEnd

### События от интерфейса

В лог попадают только те события после которых происходит переход на другой вид, или показ невидимых элементов.

Формат записи: yyyy-MM-dd hh:mm:ss Touch/Swipe идентификатор\_элемента\_интерфейса\_на\_котором\_происходит\_событие [дополнительная\_информация\_которой\_может\_не\_быть\_и\_которую\_не\_придумали\_еще]

События от интерфейса могут быть следующий видов:

1. Touch – аналог «onclick» - событие кратковременного касания после которого происходит переход на другой вид
2. Swipe – событие движения пальца по экрану.

### События показа нового вида

Формат записи: yyyy-MM-dd hh:mm:ss идентификатор\_вида [дополнительная\_информация\_которой\_может\_не\_быть\_и\_которую\_не\_придумали\_еще]

# Загрузка изображений

Приложение от сервиса получает только ссылки на изображения.

Приложение должно отправлять запрос на изображения прибавляя к ссылке идентификатор сессии.

Формат получаемой ссылки: <http://someurl.com/getimage.aspx?id=1&token>=

Формат ссылки для запроса из приложения <http://someurl.com/getimage.aspx?id=1&token>=идентификатор\_сессии.

# Интерфейс

## Главный вид приложения

Главный вид приложения – вид который показывается сразу после запуска приложения или после нажатия на кнопку «домой»

Схематичный вид представлен на рисунке 1.

Перечень элементов управления главного указан в таблице 1.

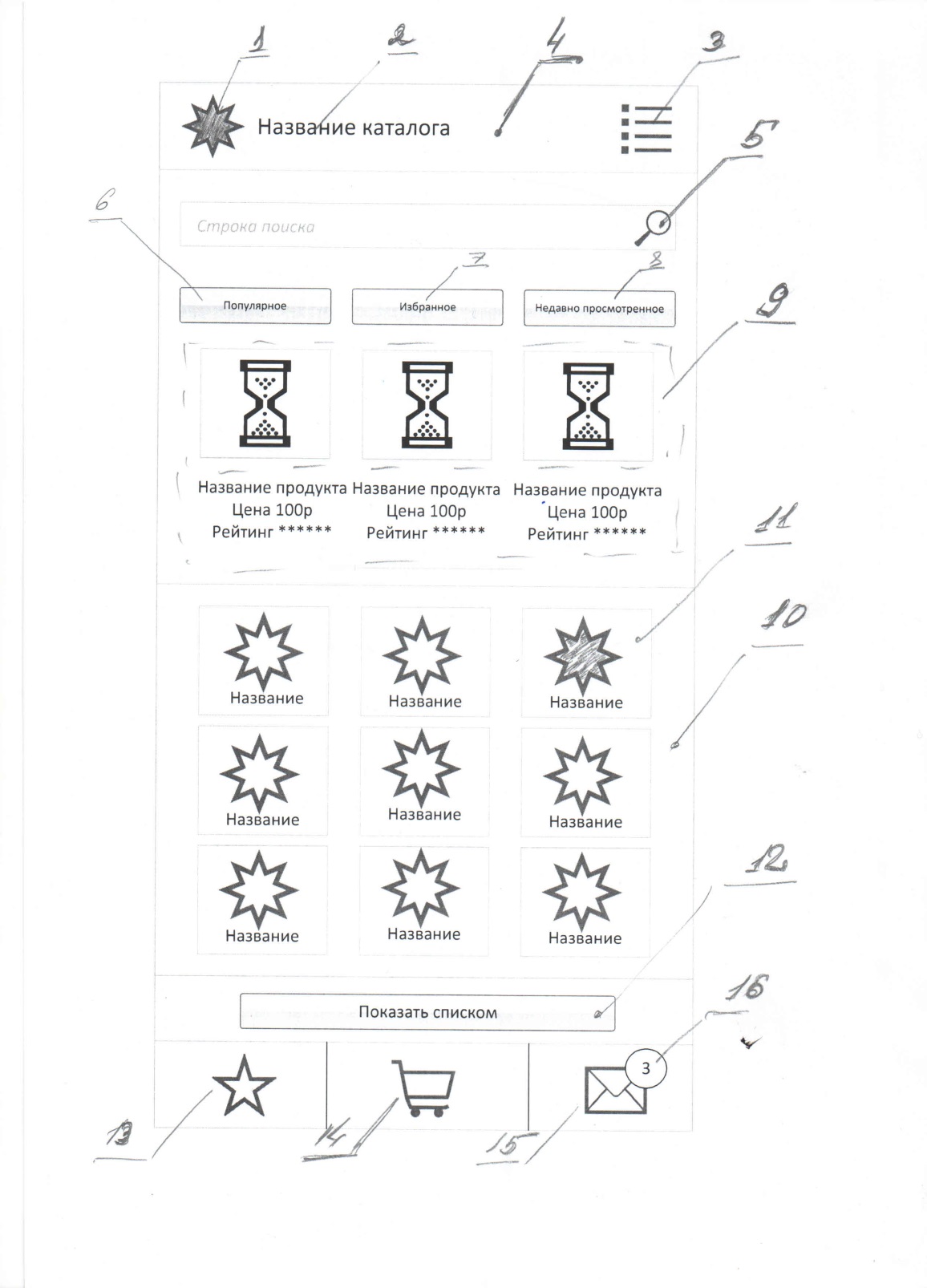


Рисунок 1

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п.п** | **Описание. Действия** | **Реагирует на события** | **Идентификатор для лога метрики** |
| 1 | Логотип каталога. URL изображения для загрузки можно получить из Catalog.LogoUrl | Нет | Нет |
| 2 | Наименование каталога | Нет | Нет |
| 3 | Кнопка перехода к виду списку доступных каталогов | Да |  |
| 4 | Блок заголовка (header) | Нет | Нет |
| 5 | Строка для поиска товаров по ключевым словам.  При наборе запроса должен появляться список подсказок. | Да |  |
| 6 | Кнопка получения популярных товаров.  Имеет два состояния: активна/неактивна.  По умолчанию активна.  Переходит в неактивное состояние если нажаты кнопки 7 или 8  Если активна, то в блоке 9 показываются популярные товары | Да |  |
| 7 | Кнопка получения избранных товаров.  Имеет два состояния: активна/неактивна.  По умолчанию - неактивна.  Переходит в неактивное состояние если нажаты кнопки 6 или 8  Если активна, то в блоке 9 показываются избранные пользователем товары | Да |  |
| 8 | Кнопка получения последних просмотренных товаров.  Имеет два состояния: активна/неактивна.  По умолчанию - неактивна.  Переходит в неактивное состояние если нажаты кнопки 6 или 7  Если активна, то в блоке 9 показываются последние просмотренные товары | Да |  |
| 9 | Блок показа избранных/популярных/последних просмотренных товаров. Блок содержит изображения товар с их наименование , ценой и рейтингом.  В блок загружаются до 10 товаров  Видимых на экране 3-4 в зависимости от конечного дизайна.  Блок должен иметь горизонтальную прокрутку.  Внизу блока должен быть индикатор прокрутки.  При нажатии на изображение товара или на текст под товаром должен происходить переход вид товара. | Да | Нет |
| 10 | Блок содержащий и изображения (иконки) разделов каталога. Построение блока происходит из локальной БД приложения.  В блок попадают только те разделы, которые у которых свойство CatalogNode.IsMajor = true.  Иконки получаются из свойства CatalogNode.Icon  В конечно дизайне будет два ряда иконок, а не три как показано на схеме.  Блок должен иметь горизонтальную прокрутку.  Внизу блока должен быть индикатор прокрутки. | Да |  |
| 11 | Блок-кнопка с изображением иконки раздела каталога.  При нажатии происходит переход на вид поиска по разделу каталога | Да |  |
| 12 | Кнопка показать списком  Может содержать две надписи «Показать списком» или «Показать все»  При нажатии происходит переход к виду всех разделов списком. | Да |  |
| 13 | Кнопка перехода к виду избранных товаров | Да |  |
| 14 | Кнопка перехода к корзине | Да |  |
| 15 | Кнопка перехода списку сообщений. Должна содержать индикатор непрочитанных сообщений. | да |  |
| 16 | Индикатор непрочитанных сообщений. | Нет | Нет |

### При активации главного вида должны быть выполнены следующие действия:

1. Из локального кеша приложения должны быть получены списки популярных, избранных и последних просмотренных товаров.
2. Отправлены запросы веб сервису на получение актуальных списков популярных, избранных и последних просмотренных товаров.
3. После получения ответа списки должны быть закешированны и отображены в интерфейсе.
4. Для списка популярных загружаются все изображения
5. Для списков избранных и последних просмотренных только первые три. Дозагрузка изображений происходит при активации соответствующего списка.

Выпадающий список подсказок при вводе в строку поиска.

Для IOS

Реализация должна быть такая же ка и в приложении «Заметки» или «Сафари»

При начале ввода в поисковую строку, текущий вид заменяется другим, где строка поиска находится на том же месте, а все место ниже заполняется списком, в который попадают подсказки. Может быть это не вид меняется, а поверх всего просто накладывается список на весь экран...надо разобраться…но это сплошь и рядом в АйФоне, и значит пользователи так привыкли.

Логика работы

Список подсказок должен строится из двух частей:

1. Истории поисковых запросов хранящихся на телефоне.
2. Из подсказок полученных от сервера

Получение подсказок от сервера

При вводе поискового запроса замеряется интервал между каждым новым символом. Если интервал достиг заданного значения (т.е. пользователь сделал небольшую паузу при вводе) – отправлять запрос на сервер, для получения списка подсказок.

Запрос на сервер отправляется заданный в настройках приложения адрес, и получает ответ в виде XML.

Формат запроса для получения: <http://someurl.com/suggest.aspx?catalogId=ИДЕНТИФИКАТОР_КАТАЛОГА&request=ПОИСКОВЫЙ_ЗАПРОС>

# Перечень методов веб-сервиса, на которые есть ссылки в документе

1. GetCatalog();
2. GetCatalogNodes();
3. SaveMetrics();
4. GetInterfaceSettings();
5. GetFavorite();
6. GetPopular();
7. GetLastVisited();