<Easy Go>

Software Requirements Specification

<1.0>

<09.10.16>

<Anastasiya Fedorenko>

Lead Software Engineer

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| <date> | <Version 1> | <Your Name> | <First Revision> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | <Your Name> | Lead Software Eng. |  |
|  | A. David McKinnon | Instructor, CptS 322 |  |
|  |  |  |  |

**Table of Contents**

**REVISION HISTORY**

**DOCUMENT APPROVAL**

**1. INTRODUCTION**

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

**2. GENERAL DESCRIPTION**

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 General Constraints

2.5 Assumptions and Dependencies

**3. SPECIFIC REQUIREMENTS**

3.1 External Interface Requirements

*3.1.1 User Interfaces*

*3.1.2 Hardware Interfaces*

*3.1.3 Software Interfaces*

*3.1.4 Communications Interfaces*

3.2 Functional Requirements

*3.2.1 <Functional Requirement or Feature #1>*

*3.2.2 <Functional Requirement or Feature #2>*

3.3 Use Cases

*3.3.1 Use Case #1*

*3.3.2 Use Case #2*

3.4 Classes / Objects

*3.4.1 <Class / Object #1>*

*3.4.2 <Class / Object #2>*

3.5 Non-Functional Requirements

*3.5.1 Performance*

*3.5.2 Reliability*

*3.5.3 Availability*

*3.5.4 Security*

*3.5.5 Maintainability*

*3.5.6 Portability*

3.6 Inverse Requirements

3.7 Design Constraints

3.8 Logical Database Requirements

3.9 Other Requirements

**4. ANALYSIS MODELS**

4.1 Sequence Diagrams

4.3 Data Flow Diagrams (DFD)

4.2 State-Transition Diagrams (STD)

**5. CHANGE MANAGEMENT PROCESS**

**A. APPENDICES**

A.1 Appendix 1

A.2 Appendix 2

**1. Introduction**

**1.1 Purpose**

Данный SRS документ описывает функциональные и нефункциональные требования к версии 1.0 приложения Easy Go. Документ предназначен для использования членами команды, которые будут реализовывать, и проверять корректность функционирования системы.

**1.2 Scope**

Easy Go позволит пользователям создавать свои карты кафе, Wi-Fi, автоматов воды и т. д. Пользователь будет иметь возможность выбирать характеристики, которые описывают сущность (цена, график работы, пароль и т. д.), и создавать карту, на которые другие пользователи могут наносить свои точки.

**1.3 Definitions, Acronyms, and Abbreviations**

API - Application Program Interface

Android SDK - Software Development Kit.

**1.4 References**

User Interface Guidelines, https://developer.android.com/guide/practices/ui\_guidelines/index.html

**1.5 Overview**

В остальной части документа описываются основные функциональные и нефункциональные требования к разрабатываемому приложению Easy Go.

Таким образом, раздел 2 - общее описание продукта и требований к нему (здесь можно прочитать об общих ограничениях, накладываемых на продукт, предположениях и зависимостях, общих функциях и перспективе продукта.

Раздел 3 - специфические требования (основные требования пользовательского, программного, коммуникационного и аппаратного интерфейсов, требования, касающиеся надежности, поддерживаемости, переносимости, подробное описание функциональных требований.

**2. General Description**

**2.1 Product Perspective**

Планируется разработка приложения, которое давало бы возможность отмечать на картах кафе, автоматы воды, киоски и другое. Главной целью приложения будет не нанесение метки, как у его аналогов, а то, что к нанесенным меткам можно будет добавлять разные характеристики, по которым будет возможность осуществлять поиск, сортировки и другие операции.

**2.2 Product Functions**

-Возможность отмечать на картах кафе, Wi-Fi, автоматы воды, киоски.

- Возможность выбирать характеристики, которые описывают сущность.

- Возможность осуществлять поиск и сортировку меток по пользовательским характеристикам.

- Возможность редактирования меток.

- Поддержка английского языка.

**2.3 User Characteristics**

Администратор – имеет возможность удалять созданные карты и управлять ними.

Зарегистрированный пользователь – имеет возможность создавать свои карты, выбирать характеристики, которые описывают добавленные на карту сущности, добавлять точки на карты других пользователей, редактировать поставленные метки. Также имеет возможность осуществлять поиск и сортировку меток по определенных характеристиках.

Незарегистрированный пользователь – имеет возможность просматривать карты других пользователей, осуществлять поиск и сортировку меток по определенных характеристиках.

**2.4 General Constraints**

Серверная часть должна быть разработана с использованием языка программирования JAVA. Клиентская часть должна быть реализована на JAVA с использованием Android SDK и Google Maps API.

**2.5 Assumptions and Dependencies**

Мы предполагаем, что ежедневно с помощью данного приложения будет создаваться от 10 до 100 уникальных пользовательских карт. Мы предполагаем, что одновременное количество активных пользователей будет составлять 1-1000 человек.

Карта считается созданной, если на ней есть хотя бы одна метка. Карта видна другим пользователям, если на ней есть хотя бы 10 меток.

Работоспособность приложения зависит от правильного функционирования Google Maps API, а также от наличия интернет соединения.

Системные требования:

* Android 4.0 или выше;
* Оперативная память 256 мб.
* Процессор с тактовой частотой 1 ГГц.

**3. Specific Requirements**

*This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation, and testing.*

*Each requirement in this section should be:*

* *Correct*
* *Traceable (both forward and backward to prior/future artifacts)*
* *Unambiguous*
* *Verifiable (i.e., testable)*
* *Prioritized (with respect to importance and/or stability)*
* *Complete*
* *Consistent*
* *Uniquely identifiable (usually via numbering like 3.4.5.6)*

*Attention should be paid to the carefuly organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.*

**3.1 External Interface Requirements**

**3.1.1 User Interfaces**

* Интерфейс пользователя должен полностью соответствовать User Interface Guidelines.
* Каждая страница будет оснащена кнопкой помощи.

**3.1.2 Hardware Interfaces**

Система не имеет аппаратных интерфейсов.

**3.1.3 Software Interfaces**

* Клиент будет получать из сервера данные о картах и метках.
* Клиент будет запрашивать у Google Maps API базовые изображения карт.
* Клиент будет отправлять на сервер данные о добавленных и отредактированных метках.

**3.1.4 Communications Interfaces**

**3.2 Functional Requirements**

3.2.1 Добавление карт

3.2.1.1 Introduction

Система должна позволять пользователю добавить новую карту с выбранными атрибутами

3.2.1.2 Inputs

Название карты, атрибуты

3.2.1.3 Processing

Пользователь должен перейти в режим добавления карт, ввести в систему название новой карты и выбрать для каждого атрибута его имя и тип.

3.2.1.4 Outputs

Появляется новая карта, доступная для всех остальных пользователей системы.

3.2.1.5 Error Handling

При попытке создать карту, которая дублирует одну из имеющихся, пользователю будет отказано в операции.

3.2.2 Добавление меток

3.2.2.1 Introduction

Система должна предоставлять возможность добавлять метки на существующую карту

3.2.2.2 Inputs

Местоположение и значения атрибутов метки

3.2.2.3 Processing

Пользователь должен выбрать карту, после этого при переходе в режим добавления метки, система считывает данные о новой метке, создает ее на соответствующей карте.

3.2.2.4 Outputs

На карте появляется новая метка.

3.2.2.5 Error Handling

3.2.3 Сортировка и фильтрация

3.2.3.1 Introduction

Система должна сортировать и производить фильтрацию меток на определенной карте.

3.2.3.2 Inputs

Параметры сортировки и фильтрации на основе атрибутов текущей карты

3.2.3.3 Processing

Пользователь выбирает желаемую карту и параметры, по которым будут осуществляться фильтрация и сортировка меток

3.2.3.4 Outputs

На выходе система выдает список меток в нужном порядке и количестве

3.2.3.5 Error Handling

3.2.4 Изменение и удаление меток

3.2.4.1 Introduction

Система должна предоставлять возможность отредактировать или убрать определенную метку

3.2.4.2 Inputs

Метка, параметры, которые нужно изменить

3.2.4.3 Processing

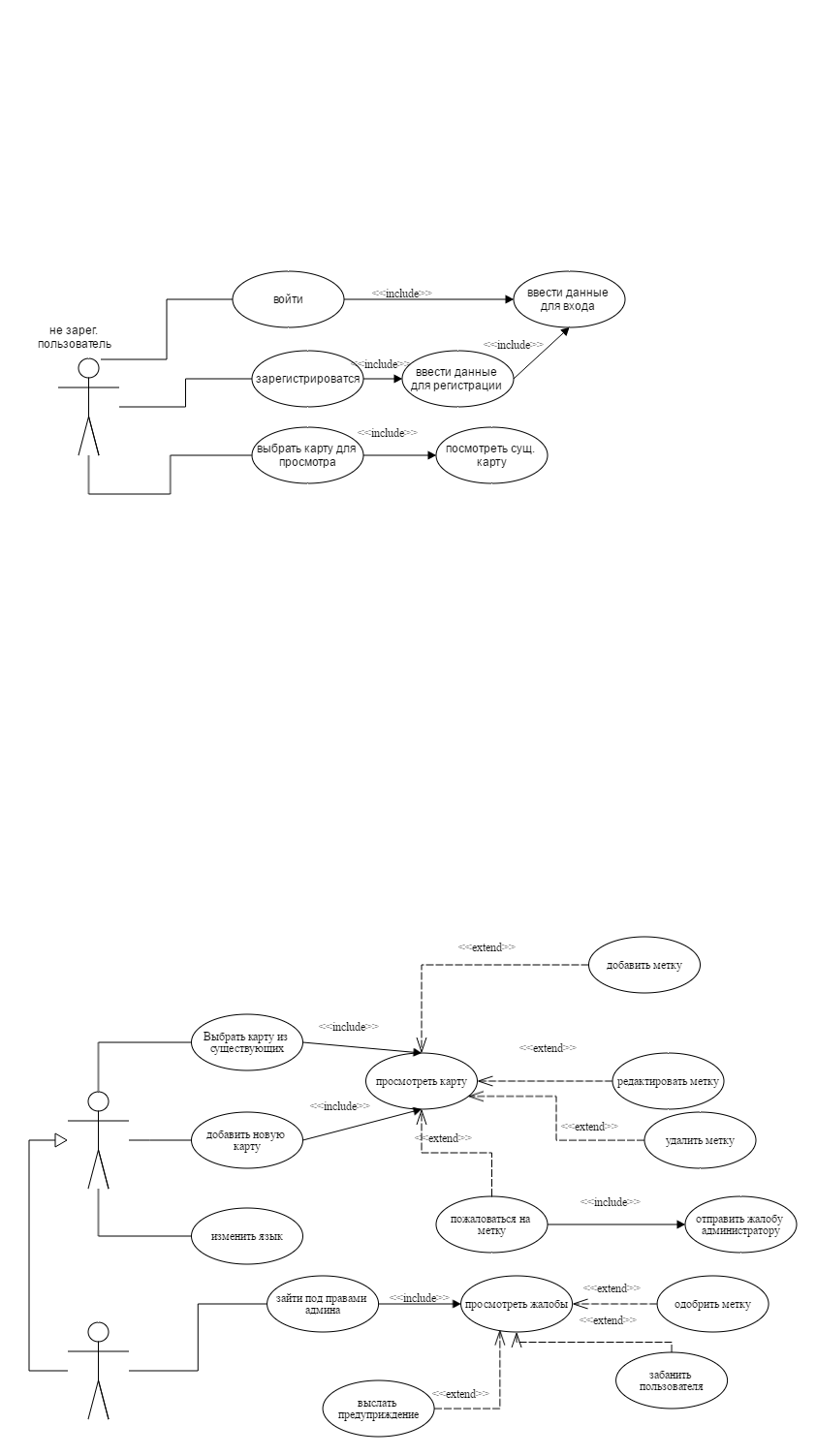
Пользователю необходимо выбрать метку на карте и перейти в режим редактирования, где он может изменит текущие значения ее атрибутов или удалить метку с карты

3.2.4.4 Outputs

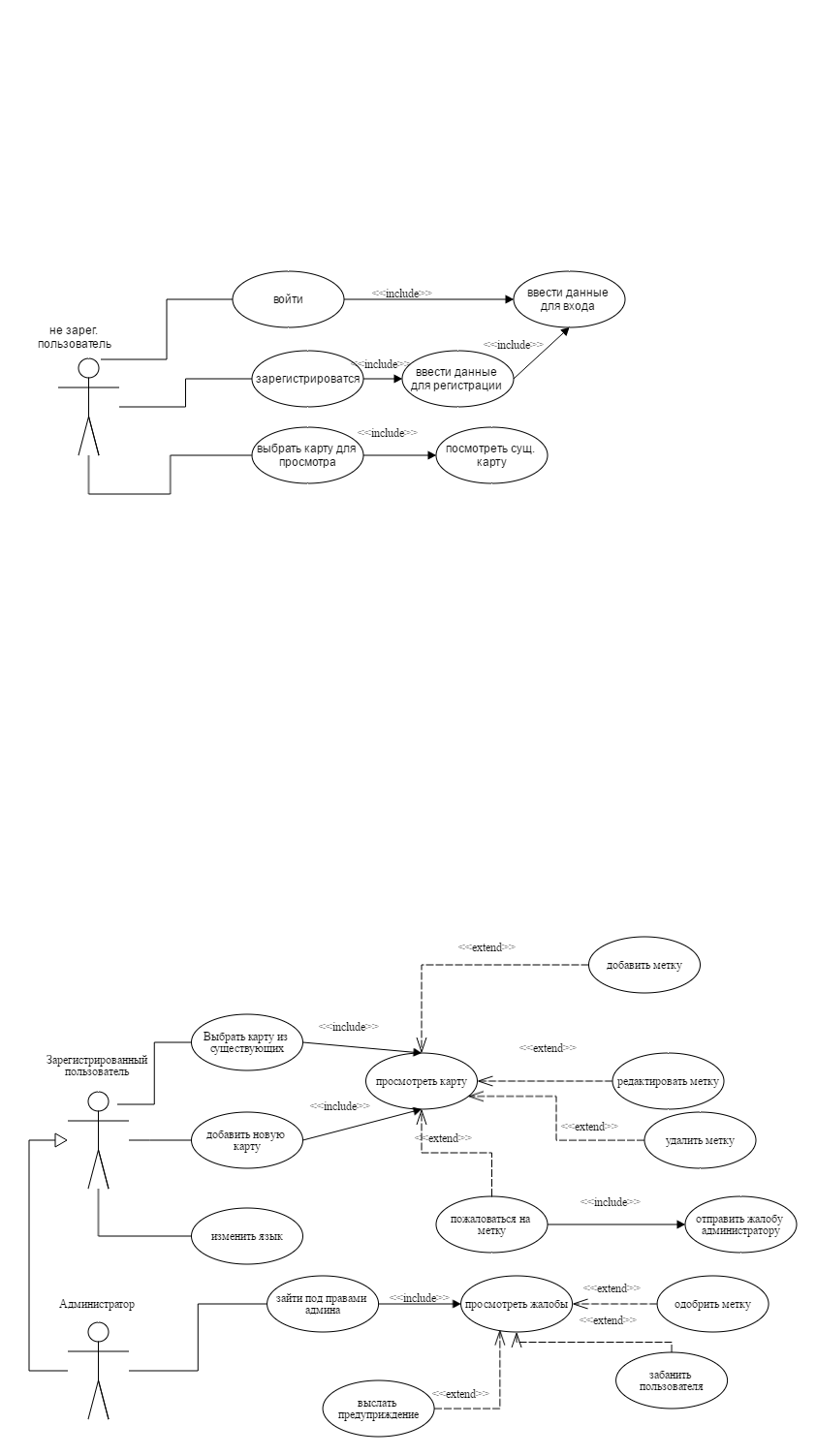
Система применяет внесенные изменения в метки

3.2.4.5 Error Handling

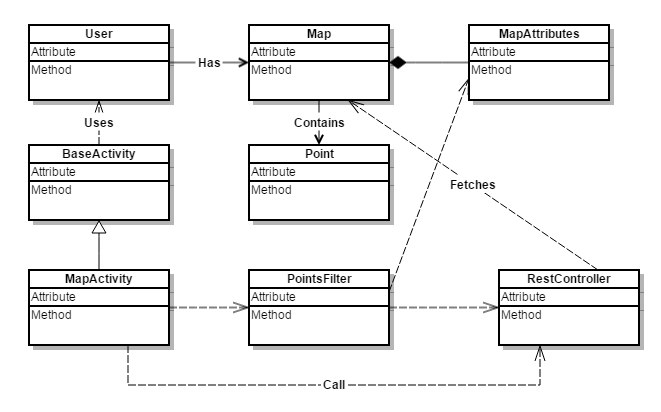
**3.3 Use-cases**

3.3.1 Use Case #1 

3.3.2 Use Case #2



**3.4 Classes/Objects**



**3.5 Non-Functional Requirements**

*Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).*

**3.5.1 Performance**

Система должна выдерживать нагрузку в 100 000 активных пользователей.

Ответ на пользовательский запрос должен формироваться не дольше чем 2 секунды.

**3.5.2 Reliability**

Система должна безотказно работать при пиковых нагрузках - 10 000 пользователей.

**3.5.3 Availability**

Система должна быть доступна 24/7/365 в любой точке земного шара.

**3.5.4 Security**

Осуществлять работу с картами в режиме редактирования могут только зарегистрированные пользователи.

**3.5.5 Maintainability**

Система должна предоставлять возможность обновления серверного ПО с минимальными затратами, чтобы не прерывать работу главного сервера и предоставить непрерывную работу с ним. Клиентское приложение будет выгружаться на Play Market и пользователи в любой момент могут получить свежую версию приложения.

**3.5.6 Portability**

Приложение будет размещено в официальном магазине приложений и будет доступно для скачивания. Сервис будет доступный для всех версий Android начиная с версии 4.0.

**3.6 Inverse Requirements**

*State any \*useful\* inverse requirements.*

**3.7 Design Constraints**

Серверная часть должна быть разработана с использованием языка программирования JAVA. Клиентская часть должна быть реализована на JAVA с использованием Android SDK и Google Maps API.

Для использования данного приложения клиент должен иметь устройство, управляемой ОС Android версии не меньше чем 4.0. и иметь операционную память не меньше чем 256 мб. Для корректного функционирования ПО пользователь должен иметь подключение к сети Интернет.

**4. Analysis Models**

**4.1 Sequence Diagrams**

