МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования   
«САНКТ–ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой №\_\_\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

БАКАЛАВРСКАЯ РАБОТА

|  |  |
| --- | --- |
| на тему |  |
|  | |
|  | |

|  |  |
| --- | --- |
| выполнена |  |
| фамилия, имя, отчество студента в творительном падеже | |

|  |  |  |  |
| --- | --- | --- | --- |
| по направлению подготовки | .03. |  |  |
|  | код |  | наименование направления |
|  | | | |
| наименование направления | | | |
| направленности |  |  |  |
|  | код |  | наименование направленности |
|  | | | |
| наименование направленности | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент группы № | 4831 |  |  |  | К.А. Корнющенков |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Руководитель

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

Оглавление

[ВВЕДЕНИЕ 5](#_Toc103004117)

[1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc103004118)

[1.1 Наименование программного изделия 6](#_Toc103004119)

[1.2 Назначение и область применения 6](#_Toc103004120)

[2. Обоснование и описание выбранных технологий разработки 8](#_Toc103004121)

[2.1 Язык программирования Swift 8](#_Toc103004122)

[2.1.1 История создания 8](#_Toc103004123)

[2.1.2 Особенности Swift 8](#_Toc103004124)

[2.2 Облачная база данных Firebase 9](#_Toc103004125)

[2.2.1 Преимущества Firebase 9](#_Toc103004126)

[2.2.2 Недостатки Firebase 9](#_Toc103004127)

[2.3 Машинное обучение 10](#_Toc103004128)

[2.3.1 CoreML 10](#_Toc103004129)

[2.3.2 Vision 11](#_Toc103004130)

[5. Что-то 3 11](#_Toc103004131)

[ЗАКЛЮЧЕНИЕ 12](#_Toc103004132)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 13](#_Toc103004133)

[ПРИЛОЖЕНИЕ 1 - Листинг программы 14](#_Toc103004134)

[ПРИЛОЖЕНИЕ 2 - Руководство пользователя 15](#_Toc103004135)

[ПРИЛОЖЕНИЕ 3 - Руководство программиста 16](#_Toc103004136)

* **Введение** - это краткое изложение мотивации написания диплома и поставленной задачи. Шаблон содержания: актуальность темы – обзор аналогов – вывод о необходимости разработки новой программы. Рекомендуемая структура введения должна содержать порядка двух страниц с описанием проблематики, затем анализ имеющихся аналогов и вывод о необходимости разработки нового программного обеспечения. Этот последний пункт обязателен - иначе непонятно, зачем городить огород и создавать что-то новое, если можно взять уже имеющийся продукт.
* **Первый раздел** имеет смысл посвятить расширенному описанию предметной области. Всегда нужно помнить, что Вас готовили не только, как кодировщика, но и как проектировщика и аналитика. Старайтесь показать, что Вы способны разобраться в предметной области, не связанной с IT. В этом же разделе имеет смысл расширить постановку задачи и описать требования к системе.
* **Второй раздел** имеет смысл посвятить проекту приложения. В проекте используйте диаграммы UML и ER. Описывая реализацию, можно приводить НЕБОЛЬШИЕ осмысленные фрагменты кода с пояснениями, что они делают
* **Третий раздел** обычно содержит в себе описание экранных форм приложения со скриншотами и руководством пользователя.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ВВЕДЕНИЕ

1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

2. РАЗРАБОТКА АЛГОРИТМА ПОСТРОЕНИЯ ГИБКОЙ ДИАГНОСТИЧЕСКОЙ ПРОЦЕДУРЫ АНАЛИЗА ТЕХНИЧЕСКОГО СОСТОЯНИЯ ОБЪЕКТА, ОПТИМАЛЬНОЙ ПО КРИТЕРИЮ МАКСИМУМА ЦЕННОСТИ ИНФОРМАЦИИ

2.1. Схема алгоритма построения гибкой диагностической процедуры анализа технического состояния объекта, оптимальной по критерию максимума ценности информации при использовании дискретных диагностических признаков

2.2. Схема алгоритма построения гибкой диагностической процедуры анализа технического состояния объекта, оптимальной по критерию максимума ценности информации при использовании непрерывных диагностических признаков

3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ГИБКОЙ ДИАГНОСТИЧЕСКОЙ ПРОЦЕДУРЫ АНАЛИЗА ТЕХНИЧЕСКОГО СОСТОЯНИЯ ОБЪЕКТА, ОПТИМАЛЬНОЙ ПО КРИТЕРИЮ МАКСИМУМА ЦЕННОСТИ ИНФОРМАЦИИ

3.1. Выбор языка программирования и среды разработки

3.2. Обоснование и описание выбираемых методов и алгоритмов разработки

3.3. Описание структур данных

3.4. Разработка интерфейса пользователя

3.5. Анализ эффективности разработанной программы

ЗАКЛЮЧЕНИЕ

# ВВЕДЕНИЕ

# ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Наименование программного изделия

Приложение по распознаванию достопримечательностей Санкт-Петербурга.

## 1.2 Назначение и область применения

«Приложение по распознаванию достопримечательностей Санкт-Петербурга» может использоваться туристами, жителями города и прочими заинтересованными лицами для решения следующих задачи:

1. Распознавать и определять достопримечательность города по фотографии, которая выбрана пользователем мобильного приложения из галереи мобильного устройства.
2. Распознавать и определять достопримечательность города по наведению главной камеры мобильного девайса на исторический объект.
3. Сокращение времени опознавания достопримечательности.
4. Получение других фотографий для распознанной достопримечательности.
5. Получение краткой исторической справки, распознанного исторического объекта.
6. Отобразить найденную достопримечательность на карте (toDo добавить)

# Обоснование и описание выбранных технологий разработки

## 2.1 Язык программирования Swift

Swift – язык программирования общего назначения, который создан прежде всего для разработки приложения на IOS и macOS.

### 2.1.1 История создания

Язык программирования был представлен общественности в 2014 года на ежегодной всемирной конференции для разработчиков на платформах Apple – Worldwide Developers Conference, которая проводится ежегодно в Apple Park.

Swift был заложен в платформе Next, поэтому в языке можно часто встретить базовые классы, особенно при взаимодействии с Obj-c, которые начинаются с приставки NS, такие как  NSString, NSNumber, NSObject и другие. Разработкой текущий версии языка занимались сотрудника Apple еще с 2010 года. Данный язык не имеет так такового одного «прородителя». Создатели взяли лучшее от разных языков, например, Obj-c, Haskell, Python, С#, С и других.

### 2.1.2 Особенности Swift

Swift является молодым языком программирования, но при этом активно развивается, в нем учтены многие минусы, которые свойственны более старым языкам программирования. Например, Swift хорошо подходит в качестве первого языка программирования. При разработке ЯП создатели преследовали такие цели, как простота изучения и ускорения процесса разработки. Данный язык получил простой и лаконичный синтаксис.

Один из главных преимуществ Swift – высокая производительность. Согласно тестам на производительность язык в 3 раза быстрее своего главного конкурента Obj-c и почти в 8 раз быстрее, чем Python. Помимо скорости Swift обладает хорошим выбором современных библиотек, которые позволяют решать разнообразные задачи начиная от написания базовых клиент-серверных приложения и заканчивания написанием ML и AR.

Но, к сожалению, у языка есть и недостатки. Одним из таких является взаимодействие с Obj-c, для которого используется специальный «мост», который сильно тормозит процесс сборки приложения. Так проект с каким-нибудь крупным приложением спокойно может собираться на 30-40 минут на современных компьютерах.

## 2.2 Облачная база данных Firebase

Firebase – облачная база данных, которая позволяет хранить и получать сохраненную информацию, а также предоставляет удобный интерфейс для взаимодействия с данными.

### 2.2.1 Преимущества Firebase

Главные плюс использования Firebase – гибкость и скорость деплоя в проекте. Также сервис предоставляет обширный спектр продуктов, которые легко между собой взаимодействуют и могут понадобиться в процессе разработки.

Ниже приведен список основных услуг, которые предоставляет Firebase:

1. Firebase Authentication – сервис, который позволяет быстро внедрить аутентификации и регистрации в приложении.
2. Firebase RealtimeDatabase – сервис для хранения и синхронизация данных в реальном времени, документно-ориентированная база данных.
3. Firebase Storage – база данных для хранения различных пользовательских файлов, документов.
4. Firebase Crashlytics – сервис, который позволяет разработчикам диагностировать и устранять проблемы в приложении.

### 2.2.2 Недостатки Firebase

К недостаткам данного сервиса можно отнести скорость и сложность запросов к БД, показатели, которые сильно влияют на производительность в больших клиент-серверных приложениях. Также при выборе Firebase на крупном проекте следует помнить о стоимости услуг, которые предоставляет поставщик.

## 2.3 Машинное обучение

Искусственный интеллект – способность машины имитировать человеческие действия и мысли.

Машинное обучение – подмножество искусственного интеллекта, которое обучает машины выполнять определенные задачи. Например, можно использовать машинное обучение для распознавание объектов на изображениях или перевода текста с одного языка на другой.

Глубокое обучение – один из методов обучение машины. Технология имитирует человеческий мозг, который состоит из нейронов, организованных в сеть. Глубокое обучение обучает искусственную нейронную сеть на основа предоставленных данных.

### 2.3.1 CoreML

CoreML – фреймворк компании Apple, который она представила на конференции WWDC в 2017 году. На его основа компания сама реализовывает такие продукты, как Siri, Camera, QuickType и другие.

CoreML - позволяет упростить интеграцию машинного обучения в приложение и создавать сложный функционал с помощью пару строк кода.

#### 2.3.1.1 Достоинства CoreML

Одни из базовых возможностей CoreML:

1. Распознавание изображений в реальном времени
2. Предиктивный ввод текста
3. Распознавание образов
4. Распознавание рукописного текста
5. Ранжирование поиска
6. Распознавание лиц
7. Идентификация голоса
8. Определение музыки
9. Реферирование текста

Core ML позволяет легко импортировать в ваше приложение различные алгоритмы машинного обучения, такие как: tree ensembles, SVMs и generalized linear models. Он использует низкоуровневые технологии, такие как Metal, Accelerate и BNNS. Результаты вычислений происходят почти мгновенно.

#### 2.3.1.2 Недостатки CoreML

К сожалению, решение от Apple далеко не идеально. Для обучения разработчик должен загружать данные в определенном формате. Также модель должна храниться на клиента, что влияет сильно влияет на исходных размер приложение. А также модель никак не шифруется и не сжимается, что негативно сказывается на безопасности.

### 2.3.2 Vision

Vision – является надстройкой над CoreML, которая позволяет применять алгоритмы компьютерного зрения для выполнения различных задач на входных изображениях и видео.

Платформа Vision выполняет распознавание ориентиров лица и лица, распознавание текста, распознавание штрих-кодов, регистрацию изображений и отслеживание общих функций. Vision также позволяет использовать пользовательские модели Core ML для таких задач, как классификация или обнаружение объектов.

# Что-то 3

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. ISO/IEC/IEEE 29148:2018 International Standard — Systems and software engineering — Life cycle processes — Requirements engineering
2. Виггерс, Карл. Разработка требований к программному обеспечению = Software Requirements: пер. с англ.; 3-е издание, дополненное / Карл Виггерс, Джой Битти — М.: Издательство «Русская редакция», 2014. — 736 с.: ил.
3. IEEE Recommended Practice for SRS / IEEE Std 830-1998 (Revision of IEEE Std 830-1993) [Электронный ресурс]. — University of Alaska Anchorage, 2020 — URL: <http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>
4. Writing Software Requirements Specifications (SRS) [Электронный ресурс]. — INKtopia Limited, 2020 — URL: <https://techwhirl.com/writing-software-requirements-specifications/>
5. Software Requirements Specification Helps to Protect IT Projects From Failure [Электронный ресурс]. — Belitsoft, 2004-2020 — URL: <https://belitsoft.com/php-development-services/software-requirements-specification-helps-protect-it-projects-failure>

# ПРИЛОЖЕНИЕ 1 - Листинг программы

# ПРИЛОЖЕНИЕ 2 - Руководство пользователя

# ПРИЛОЖЕНИЕ 3 - Руководство программиста