

Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University



Bekbossynova Aisulu
Koshimbay Aiasem
Ketebay Aidana
Temirshot Assiya
Tolegenova Assylay

**Implementing mobile art gallery guide
application based on iOS/Android
Реализация мобильного руководства по
картинным галереям на базе iOS / Android
IOS / Android негізінде арт-галереяларға
мобильді нұсқаулық енгізу**

A thesis submitted for the degree of
Bachelor in Computer Systems and Software
(degree code: 5B070400)

Kaskelen, 2020

Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University
Faculty of Engineering and Natural Sciences

**Implementing mobile art gallery guide application based
on iOS/Android**

**Реализация мобильного руководства по картинным
галереям на базе iOS / Android**

**IOS / Android негізінде арт-галереяларға мобильді
нұсқаулық енгізу**

A thesis submitted for the degree of
Bachelor in Computer Systems and Software
(degree code: 5B070400)

Author: **Bekbossynova Aisulu**
Koshimbay Aiasem
Ketebay Aidana
Temirshot Assiya
Tolegenova Assylay

Supervisor: **Kariboz Darmen**

Dean of the faculty:
Assist. Prof. Bogdanchikov Andrey

Kaskelen, 2020

Abstract

Since 2009, most museums in the world began to deploy software applications in various media and guest sectors. As mobile technologies and smartphones continue to evolve and permeate all aspects of life, becoming more convenient and popular, cultural institutions have new opportunities to use these innovations to attract tourists. More and more new ways began to appear to promote their exhibitions and programs. However, not all countries keep up with innovations in the world of technology, especially in the field of art. Kazakhstan museums were taken as an example. To solve this problem and improve museum services in general, mobile applications were implemented on iOS and Android platforms, as well as an online telegram bot for visitors. This fascinating art guide will help visitors to explore the galleries in the museum and enrolls information about the gallery in 3 languages.

Based on the implementation of the application, the AGILE methodology is used. This thesis describes how the Agile approach improves the process of implementing mobile applications on iOS and Android platforms, and also online Telegram bot. The analysis showed that the Agile approach can improve the quality of mobile applications.

Keywords: Mobile applications, iOS and Android platforms, telegram bot, museum, guide

Аңдатпа

2009 жылдан бастап көптеген әлемдік мұражайлар мобильдік қосымшаларды әр түрлі бұқаралық ақпараттар және туристік салаларына енгізе бастады. Ұялы телефон мен смартфон технологиялары өмірдің барлық жақтарын жетілдіріп, ыңғайлы және танымал болуына байланысты, мәдени мекемелер туристерді тарту мақсатымен осы инновациялық қосымшаларды қолданудың жаңа мүмкіндіктеріне ие болды. Көрмелер мен бағдарламаларды жоғары деңгейге жетілдірудің жаңа тәсілдері пайда бола бастады. Дегенмен технология әлемінде, барлық мемлекеттер жаңа инновацияларды әсіресе мәдени сектор саласында жетік менгерілмеген. Мысал ретінде Қазақстанның мәдени мұражайлары алынды. Бұл мәселені шешу және жалпы мұражайдың қызмет көрсету саласын жақсарту мақсатымен, мұражай келушілері үшін iOS және Android платформаларында мобильді қосымшамен қатар онлайн Telegram бот іске қосылды. Бұл қызықты нұсқаулық мұражайдағы галереяларды зерттеуге және галерея нысандары туралы қажетті ақпаратты 3 тілде алуға көмектеседі.

Қосымшаны іске асыру негізінде AGILE әдістемесі қолданылды. Бұл тезисте Agile әдістемесі iOS және Android платформаларындағы мобильді қосымшаларды және онлайн Telegram ботын іске асыру процесін қалай жақсартатындығы туралы сипатталған. Іске асырудың нәтижесінде Agile тәсілі мобильді қосымшалардың сапасын жақсартуға мүмкіндік беретінін көрсетті.

Кілт сөздер: мобильді қосымшалар, iOS және Android платформалары, Telegram bot, мұражай, гид

Аннотация

С 2009 года большинство музеев мира начали разворачивать программные приложения в различных медиа и гостевых секторах. Поскольку мобильные технологии и смартфоны продолжают развиваться и пронизывать все стороны жизни, становясь все более удобными и популярными, у культурных учреждений появляются новые возможности использовать эти инновации для привлечения туристов. Стало появляться все больше новых способов для продвижения своих выставок и программ. Однако не все страны идут в ногу с инновациями в мире технологий, особенно в области искусства. В качестве примера были взяты казахстанские музеи. Для решения данной проблемы и улучшения музейных услуг в целом, были реализованы мобильные приложения на двух платформах iOS и Android, а также онлайн телеграмм-бот для посетителей. Этот увлекательный арт-гид поможет вам изучить галереи в музее и получить нужную информацию об объектах галереи на 3 языках.

Основываясь на реализации приложения, используется методология "Agile". В этом тезисе рассказывается о том, как Agile-подход улучшает процесс внедрения мобильных приложений на платформах iOS и Android, а также онлайн Telegram-бота. В результате реализации показано, что Agile-подход позволяет повысить качество разработки мобильных приложений.

Ключевые слова: мобильные приложения, платформы iOS и Android, Telegram бот, музей, гид.

Contents

1	Introduction	6
1.1	Purpose	6
1.2	Motivation	6
1.3	Objectives and overview to chapters	7
2	Main Body	9
2.1	Agile methodology	9
2.2	Developing telegram bot	10
2.2.1	Pythonanywhere	11
2.2.2	Connection between client and server	12
2.2.3	Database	14
2.2.4	Image search engine	15
2.2.5	Natural language Processing	19
2.2.6	Corresponding functions for commands	21
2.3	Implementing iOS and Android application	24
2.3.1	Mobile tools	24
2.3.2	Database	25
2.3.3	Image recognition	26
2.3.4	Functions for playing audio	30
2.3.5	Additional features	32
3	Conclusion	36
3.1	Future development	37
A	Appendix	38
	References	40

Chapter 1

Introduction

1.1 Purpose

Despite the world financial condition, the mobile technology market is one in all the few industries that see a steady and fast growth. Globally, there are around 205 billion mobile apps, 1.35 billion tablet apps, and rest a billion mobile apps downloaded from smartphones and tablets. (Key Mobile app Statistics, 2019).[1] It indicates an implausible chance for the software industry to leverage a growing increasing demand and maintain relationships with a rising forms of applicants across several different domains. It brings up an out sized sort of circumstances in the travel industry to draw and maintain tourists' loyalty. Tourist destinations such as museums will not neglect its potential and must find new avenues to realize the of a replacement form of tourist, becoming "Museum 2.0."

As a result of the projected growth of latest mobile technology, and the upcoming future museum projects, it is obvious that the mobile technologies will actually become a visitor guide. The Museum guide app is an interactive guide that affords guests with electronic information about the item in an exceeding museum in text and audio format.

The foremost purpose of the project is to establish a visible search algorithm that may classify similar items based on the characteristics or key-points of an input object.[2] It has targeted on exploring and introducing new technique to refine the proposed algorithm and its convergence with mobile apps and Telegram bot . The "ArtGuide" app propose to help users find and provide data regarding their museum objects of interest easily. The visual search appliance takes input in the form of keywords or graphic images and offers information of the artwork and its history in three languages in acceptable formats (audio and text).

1.2 Motivation

Special expositions at the national museum in Kazakhstan are drawing more and more visitors. Over 150,000 people have toured the middle over the past

two months. Valuable artifacts and collections found within the World Heritage Program has drawn other lovers of culture. Each of the museum's ten exhibition halls indicates numerous periods: from the new the traditional. Several of the foremost common areas are those that house the famed Golden Man, important findings from ancient burial sites and settlements.

In keeping with that data, further difficulties occur in supplying details to every guest thanks to the shortage of museum employees. In fact, most tourists are foreigners and language difference is an intimidating issue.

To attenuate these difficulties and increased interest in Kazakhstan's history is to make a web ArtGuide program for visitors that might not only ease the museum employees' activities but also allow visitors' stay in it more immersive and interesting.

The goal of the project is to create a visual search program which recovers data from image input. This search application gives details supported the algorithm which matches the picture. Such kinds of apps are frequently employed in the smartphone industry, where a user can collect details about unknown objects by taking a picture and ransacking through the utilization of the ready gadget.

Visual content is far simpler to understand than interpreting text information. It grants to change gathered information in a graphical, text, or audio format. That creates a good section of the software usable. Data is defined in a very structured and precise manner, because it can offer more detailed knowledge to the user. The usage of such software for mobile devices are limitless, despite the newest age of smartphones. By utilizing mobile devices, and also the image program algorithm will establish a variety of fields and applications. This is able to introduce a contemporary technology pattern into the forthcoming market entirely.[3]

1.3 Objectives and overview to chapters

The main focus of the appliance is predicated on Visual search. Visual search apps are those programs that have visual content like a picture, video, etc. From those categories, it's used Image computer program on applications.

The Visual Search program includes a spacious extension within the mobile industry. The Mobile Visual Search techniques specially designed to figure for mobile devices to retrieve information easily and quickly. Smartphones benefit from high-resolution images, high-speed internet connectivity, fast hardware acceleration, GPS, etc. This makes handheld devices the right candidate for visual Search applications.

The project's ending is an ArtGuide application seek for information by presenting the user data as a picture and storing it within the following manners:

- The client takes through a cell camera on a picture of the requested object.

- The image data are submitted to the server for image recognition and detection of comparable pictures.
- The server performs all the computations required to locate compatible photos and sends details back to the mobile app.

The next chapters describe the full information of all implementations of visual search on diverse platforms like iOS/Android and online Telegram bot. Ranging from the methodology that's used for development for the appliance. Agile methodology is predicated on consumer suggestions, it takes customer reviews into consideration at each level and so integrates app criteria consistent with their needs.

Chapter 2

Main Body

2.1 Agile methodology

The mobile telecommunication industry has a highly competitive, diverse, and unpredictable environment.[3] The agile method is often seen as a natural fit for the event of mobile applications. Studies have undertaken to adapt the agile architecture approach to the assembly of mobile apps that show the requirement for product engineering frameworks customized to the wants of mobile applications.[4] This has been proposed that agile methods are the safest choice to maintain various stages of the life cycle of development and more effectively address mobile device development problems.[5] There are several factors why agile approaches are chosen for mobile application implementation:

- **Reduced Risks**

The agile process uses a parallel test model to facilitate concurrently designing certain apps to line up the assorted mobile device development problems. The consistency that the tactic provides with the Agile Software Creation model scrum method enables seeing and extracting the issues in even the later stages from their origin and within the initial stages themselves.

- **Enhanced productivity**

Through teamwork, the team of mobile device creators operates for consistency on targets, which helps meet all obligations in an exceedingly specified period.

- **Build higher-quality technologies for mobile device creation**

Due to the collateral check methodology utilized in the agile mobile device development phase, it's simpler for mobile app developers to review a software prototype in real-time then hand-by-hand experience the requisite quality-based changes.

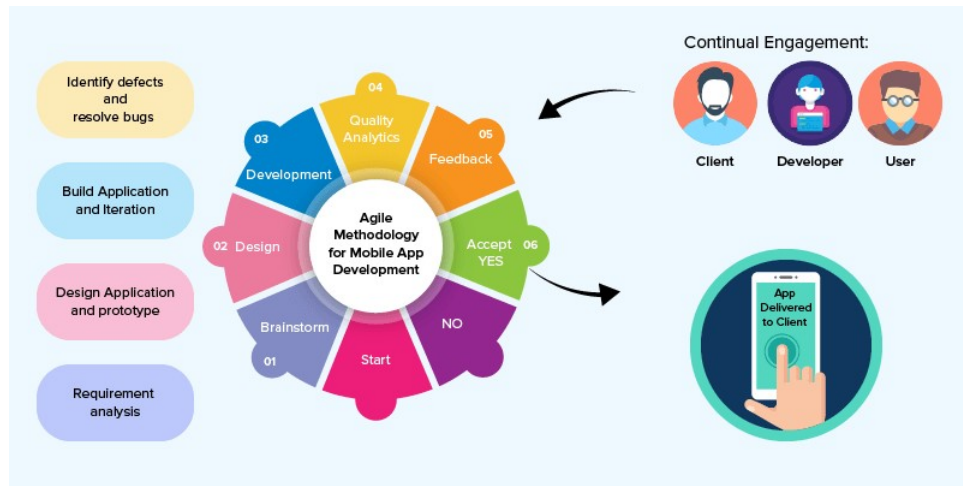


Figure 2.1: Principles of Agile methodology

For implementing a visible search on the appliance, it is divided for tiny tasks like creating a design, databases, and other functions that may be described deeply within the next chapters.

2.2 Developing telegram bot

One of the foremost pleasant highlights of mobile phones is errand individuals. It transforms into the most instrument to speak, to exchange information, and to interact with each other. As Short Message Service (SMS) is antiquated, it is popular for people to use detachment apps like WhatsApp, Blackberry Messenger (BBM) and Telegram as guide communications.

In 2014, Facebook acquired WhatsApp to play a significant position alongside its own ambassador stage within the increasingly developing information industry. After WhatsApp, lots of social networking was built, which was trying to get on par with other common IT applications within the world. One in all the late programs is called Telegram, which functions lots the identical as WhatsApp, using cryptography from start to finish to ensure mutual data.(Pinto, 2014) The company proclaims to possess over 150 million consumers and provides highlights like tweets falling to bits to ensure more prominent security. Customers may legitimately distribute, either privately meetings, or through websites.

In Telegram, the foremost latest additional feature is Chat bots. It is a small amount of a system, supported small-level AI and deep learning. Through submitting orders, customers may match with bots, and that they will react precisely to what customers need as meeting an accomplished friend. These bots could also be modified looking on client prerequisites and inserted into numerous administrations to manage, for instance, genius houses, tuned apparatuses, and other social administrations. During this article, a telegram bot helps to implement the information exchange of the museum's artifacts.[6]

2.2.1 Pythonanywhere

The goal of developing the project was to find a reliable full Python cloud environment, which might be wont to host Telegram bot without having to download software or manage a peculiar server.

PythonAnywhere is an internet Incorporated Advancement Environment (IDE) and Webhosting administration passionate about the python programming language. It gives in-program access to server-based Python and Bash Command-line interfaces, alongside a code editorial manager with syntax.

PythonAnywhere gives free starting python content facilitating. It is progressively like a server explicitly upheld for the python, that people can remotely call and oversee it from another code base. Interestingly, it bolsters versed variants of Python further. The expectation to find out and adapt to utilize its less and hence, there are container user base and bolster accessible. Likewise, it can totally send an online application made in Python frameworks as in Django,PyTorch, and any of the python libraries. At last, PythonAnywhere is that the effortless arrangement out there for executing python contents remotely on the cloud. Moreover, in the "Task" segment there are two varieties of mechanized assignment frameworks that clients can utilize. One is "Scheduled task" which is content that will run each day at the client indicated time to try and do scratching sites, or test server is running or not. The second robotized task is "Always-on tasks" which are the contents that are continued running constantly, and restarted within the event that they crash. Clients can manufacture web applications, transfer documents to run. Along these lines, PythonAnywhere is an out of this world asset for software engineers and programming groups.

The Telegram bot created using BotFather and had the first scripts. There's faced a controversy of constructing Telegram Bot to manually request for updates to the Bot Programming Interface (API) after each client's message. In order to possess for automatically updates, it was decided to enroll a WebHook to consequently be called once refreshes are accessible. This solution would avoid Bot having to fire updates frequently and also the need for a few reasonably polling mechanism within the code.

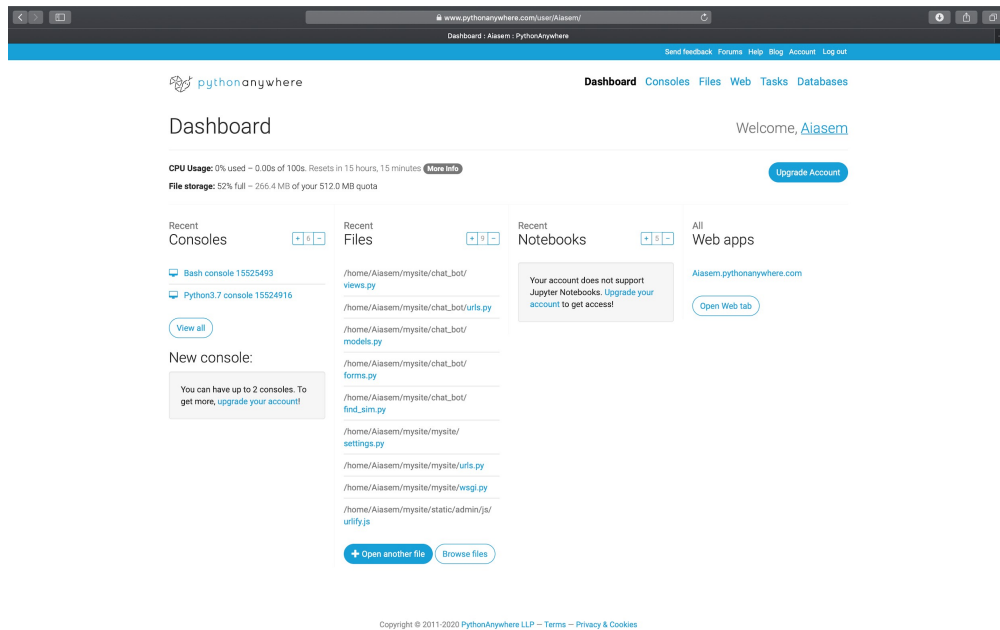


Figure 2.2: PythonAnywhere interface

2.2.2 Connection between client and server

The technology of the venture depends on Telegram, written in Python programming language, and its backend component. Since the web-customer is the undertaker's key component, one in all the foremost significant options was to settle on a suitable framework for web-application recognition. Python language includes some "full-stack" structures: Django, Grok, web2py, Giotto.

Thee appliance is written within the programming language of Python to enter the Advancement and AI process. Django framework was selected for the event change. The Telegram Bot API was accustomed to render a Telegram bot and merge it with Telegram Administrations.

Within Telegram bot, there are two advanced communications methods, there are Webhooks and get updates(Long polling).[7]Long polling could be a pull process, and push is about Webhook. Webhooks methodology is, in essence, simpler to develop, HTTP-based submission, and straightforward to reply to a message. In any event, it requires a fervent server with accessible IP while no devoted server wants an extended survey. An out sized survey methodology reduces the expense of manufacturing computerization reception. This can boost an out sized range of changes without each upgrade needing basically an API. With additional equipment, this home computerization is increasingly guarded. Webhooks are then selected in this use on the grounds that the information listed must be accessed rapidly and live day in and day out.

Using a Webhook system for receiving new updates has certain advantages:

1. Do not have to ask the bot for updates ever.

2. It does not have to include any quite polling feature within the application.
3. Besides 100 links total, webhooks don't face a fixed latency. Each bot that's quite capable of managing any load throw at it.
4. Saving some possible CPU cycle, and an improvement in latency, but such issues rely heavily on the bot's utilization pattern.

By analyzing all the benefits, the ArtGuide Bot has used the Webhook mechanism, which specifies an URL and receives incoming updates. The client must submit a HTTPS POST request to the defined address, containing a JSON-serialized update, if there's an update for the bot.

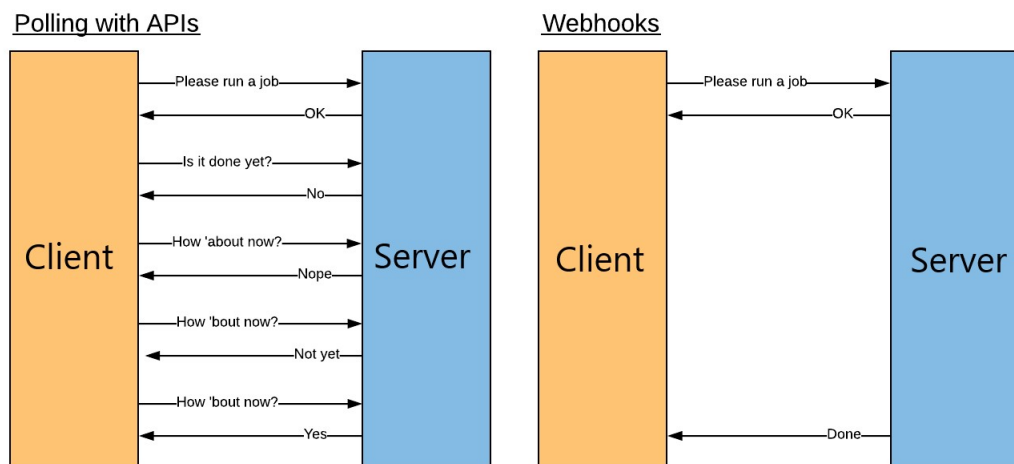


Figure 2.3: illustration of mechanisms Long Polling and Webhook

Setting of webhook for ArtGuide bot is written in `urls.py` and `view.py`

```
from . import views
from django.urls import path
from .views import TelegramBotView
from . import config

TOKEN = "bot" + config.token
urlpatterns = [
    path('', views.index, name='index'),
    path(TOKEN, TelegramBotView.as_view()),]
```

Listing 2.1: Setting Webhook

All updates from the client-side get from the URL patterns and redirects to `TelegramView()` class. As Webhook is connected, it began to watch for any updates from clients. All data from the user encoded to JSON format and updated in Bot.

```

class TelegramBotView(View):
    @method_decorator(csrf_exempt)
    def dispatch(self, request, *args, **kwargs):
        return super(TelegramBotView, self).dispatch(
            request, *args, **kwargs)

    def post(self, request):
        bot.process_new_updates(
            [telebot.types.Update.de_json(
                request.body.decode('utf-8'))])

```

Listing 2.2: TelegramBotView class

2.2.3 Database

For storing and fetching data associated with gallery objects are required databases. Django framework reinforces a wide set of Relational Database Structures (RDBS), carrying PostgreSQL, SQLite3, MariaDB/MySQL, and Oracle.

From the listed databases, SQLite offers an excellent option for designing applications that are primarily read-only or need a smaller footprint for deployment. Connection of the database is written in settings.py[8]

ArtGuide bot uses two independent tables which are not linked with foreign keys. The ArGallery table is based on the data of images in three languages (Kazakh, Russian, and English) and the second one is the information guide for bot's commands. After the setup database, it is created classes ArtGallery and InfoGuide models for storing data that are inherited from models. Model class. Using control system migration was created two independent tables in the database. The migrate command checks the configuration of INSTALLED_APPS and uses the migrations to build all the correct tables in the database specified in settings.py.

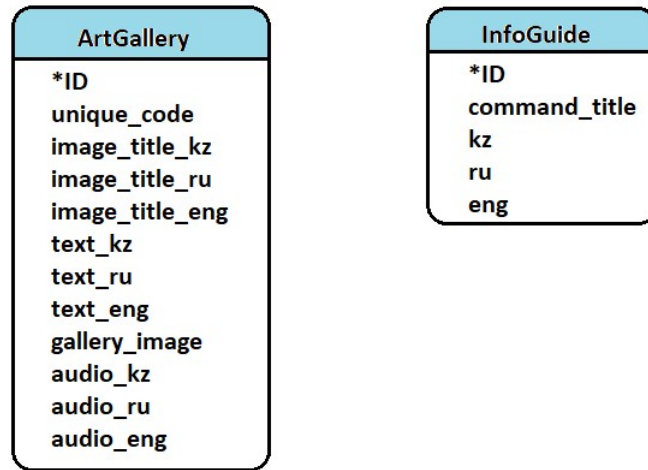


Figure 2.4: ERD of bot's table

2.2.4 Image search engine

The fundamental idea of the project is to create an online Image Search Engine, which requires an image input from the user in order to supply full information of the appropriate art object.

Highlight identification is the way toward registering the deliberation of the picture data and settling on a part choice at each picture point to test whether there is an image highlight of the given kind existing in this point. [9]

An image preparing assignments incorporates presentations; fundamental controls like editing, flipping, turning, and so forth.; picture division, order, and highlight extractions; picture reclamation; and picture acknowledgment. That is why Python is a great decision for these types of figure preparing undertakings because of its developing prevalence as a logical programming language and therefore the free accessibility of many best in class image handling instruments in its environment.

Searching through figures by the Image Search Engine is constructed by giving a contribution to the type of a picture. The coordinating pictures are discovered hooked in to the data given. This procedure is otherwise called Content-Based Image Retrieval (CBIR). In this procedure, implemented code searches key-focuses or highlights of the information picture with the other pictures' highlights in a database. This technique requires complex calculations for finding coordinating pictures. Nevertheless, Python libraries are most frequently utilized for picture control assignments. These libraries provides a simple and natural approach to change pictures and understand the fundamental information.

The goals of achieving successful results were:

1. The client snaps a figure of the ideal workmanship object through a portable camera.

2. An info picture is shipped to the server for picture handling and finding coordinating pictures.
3. The server does all calculations for finding coordinating pictures and sends back data to the client.

In this area, it has contemplated diverse picture coordinating systems and experience some picture acknowledgment strategies obsessed with a picture's highlights. To be concrete, it considered pictures include extraction techniques like SIFT and SURF in detail and locate an ideal picture coordinating method that is snappy and precise for the future application.

Dwelling in more detail on each of the above methods, it can be said:

Scale Invariant Feature Transform (SIFT) - this technique is utilized for the canny picture coordinating within which key-focuses from pictures are chosen and contrasted and one another. It chooses significant focuses on the image and afterward contrasts those focuses and the information picture. In the event that it finds that the focuses are coordinating, at that time the calculation gives pictures that are coordinating or comparative. Be that as it may, this procedure is delayed as it desires some investment to look at all the focuses. In any case, this picture coordinating system is well known as it is all-powerful to light, scale, turn.

SIFT is a component locator created by Lowe in 2004. In spite of the actual fact that SIFT has demonstrated to be effective in object acknowledgment applications, it requires a huge computational multifaceted nature which is a compelling downside particularly for continuous applications.

Accelerate Robust Feature (SURF) - is an estimation of SIFT, performs quicker than SIFT without lessening the nature of the recognized focuses. ("Speeded-up robust features(SURF)", 2008)[10] Both SIFT and SURF are subsequently founded on a descriptor and an indicator. Parallel Robust Independent Elementary Features (BRIEF) is an alternative choice for SIFT which requires less unpredictability and with practically comparable coordinating execution.

SURF and SIFT are to some degree comparable. The significant distinction in them is their methods for distinguishing highlights or key-focuses and portraying them:

- Fast key point discovery.
- Distinctive key point depiction.
- Speed-ed-up descriptor coordinating.
- Scale investigation with a steady picture size.

ORB(Oriented FAST and Rotated BRIEF) - is fundamentally a combination of FAST key point finder and BRIEF descriptor with numerous alterations

to boost the presentation. First, it utilizes FAST to find key points, at that point apply Harris corner measure to discover top N focuses among them. In any case, one issue is that FAST doesn't register the direction.[11]

Every single one of them has advantages and downsides, it relies upon the kind of pictures some calculations will recognize a larger number of highlights than another. Filter and SURF are licensed so not free for business use, while ORB is free. SIFT and SURF recognize more highlights at that point ORB, however ORB is quicker.

For image similarity detection SIFT and SURF are more suitable solutions.

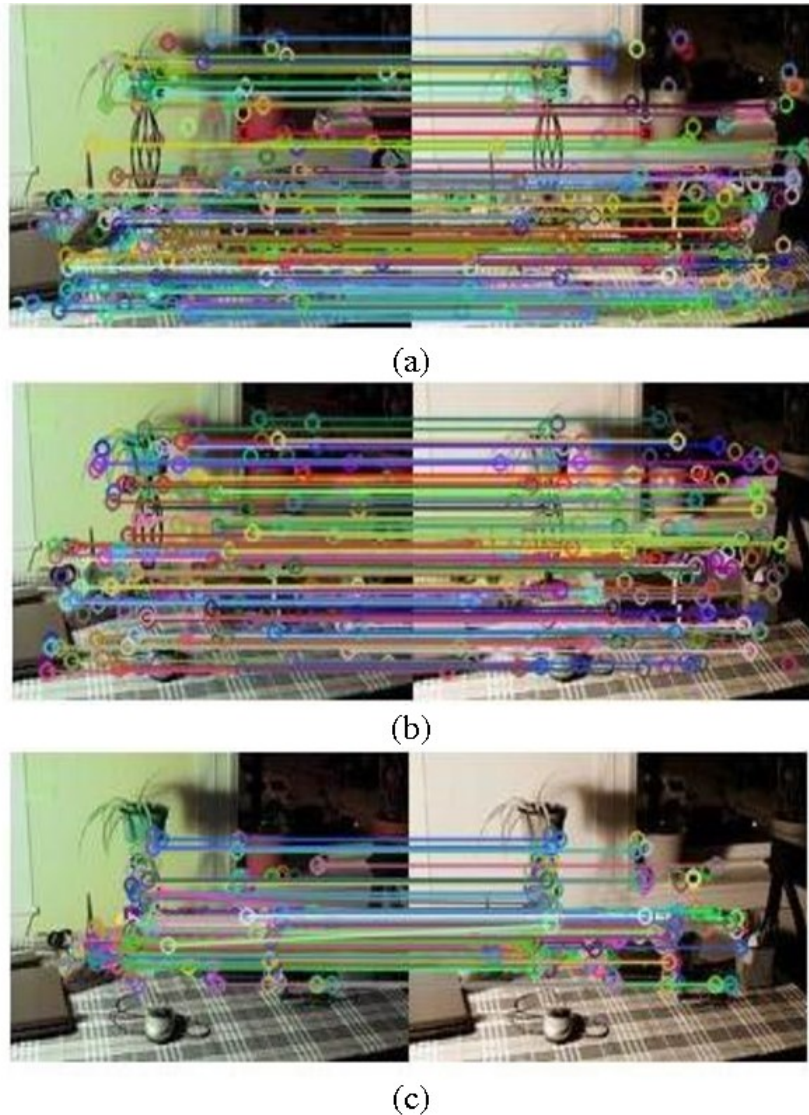


Figure 2.5: The matching of varying intensity images using (a) SIFT (b) SURF (c) ORB.

Following Figure 2.5, for images with varying intensity values, SIFT gives the best coordinating rate while ORB has the least. All things considered, computational time necessity for ORB is the least.

As a closure, SURF has as great of a performance as SIFT, yet it isn't that useful for a pivot and scale. SURF is a few times quicker than SIFT, which makes SURF a superior applicant.

	Time (sec)	Kpnts1	Kpnts2	Matches	Match rate (%)
SIFT	0.13	248	229	183	76.7
SURF	0.04	162	166	119	72.6
ORB	0.03	261	267	168	63.6

The SURF calculation is quicker and increasingly precise for picture highlight or key-points locator as looked at toward SIFT. Along these lines, the SURF algorithm for detecting local features in images was used in the project.

Figure 2.6: Comparison between algorithms.

```
def surf_sim(img_a, path_b):
    orb = cv2.xfeatures2d.SURF_create()

    img_b = cv2.imread(path_b) #read dataset image
    img_b = cv2.cvtColor(img_b, cv2.COLOR_BGR2RGB)

    # find the keypoints and descriptors with SURF
    kp_a, desc_a = orb.detectAndCompute(img_a, None)
    kp_b, desc_b = orb.detectAndCompute(img_b, None)

    # initialize the bruteforce matcher
    bf = cv2.BFMatcher()

    # match.distance is a float between {0:100}
    # lower means more similar
    matches = bf.knnMatch(desc_a, desc_b, k=2)
```

Listing 2.3: The code implementation of SURF algorithm

It is not vital that the picture caught is in an appropriate arrangement. The picture could be obscured, misshaped, sideways, and so on. Be that as it may, this is extra overhead and it could hinder the looking through the procedure. In this way, to get quicker query items it needs a calculation that is invariant to picture changes, scaling, revolutions, and staining. It utilizes the SURF intrigue focus to locate the coordinating pictures. SURF is quick and strong to picture scaling, revolution, and obscure. This entire procedure is done on the server-side.[12]

In order to get more accurate results, all the dataset and query images were converted and resized in the same color code and size respectively. Moreover, due to the fact that a picture taken from a mobile phone's camera can be shot from a long distance which could also affect the accuracy. To solve this issue, the central part of the query image will be cropped.

```

def query_process(imagepath):
    img = Image.open(imagepath)
    im = cv2.imread(imagepath)
    h = im.shape[0]
    w = im.shape[1]
    cropped_img = img.crop(((w-500)//2,(h-500)//2,
    (w+500)//2,(h+500)//2))
    image = np.asarray(cropped_img)
    return image

for image in image_list:
    image = image.resize((500,500))
    image = image.convert("RGB")
    resized_images.append(image)

```

Listing 2.4: The implementation of processing query image and dataset

2.2.5 Natural language Processing

Speech is the most well-known method for correspondence around the globe. The greatest part of the populace on the planet depends on discourse to speak with one another. Assume it is built a model and instead of a composed methodology, which needs a framework to react to discourse, it seems to be genuinely troublesome and requires an excellent deal of data to be prepared. A speech acknowledgment framework conquers this obstruction by making an interpretation of discourse to content[13]

Moreover, Python application offers a degree of intuitiveness and availability with picture processing as well as with language handling. These openness upgrades alone merit considering. Discourse acknowledgment permits the older and the genuinely and outwardly hindered to connect with best in class items and administrations rapidly and normally with no GUI required.

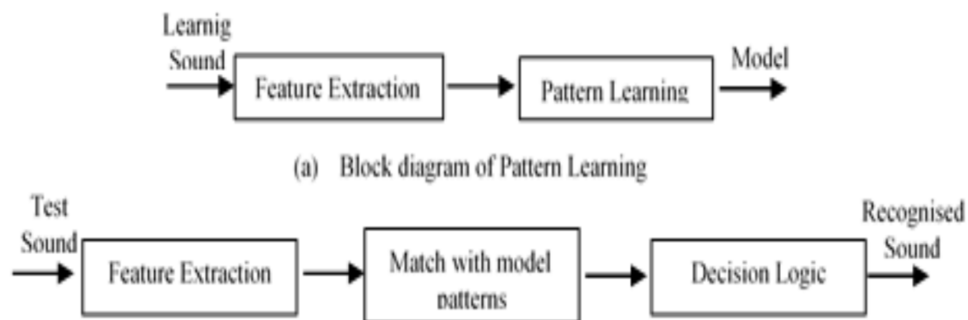


Figure 2.7: Speech recognition block diagram

Natural Language Processing(NLP) - is a part of man-made reasoning that manages the association among computers and people utilizing the common language. Its main idea is to change speech over from physical sound to an electrical sign with a microphone, and afterward to advanced information with a simple to the computerized converter. The most ease-of-use package in Python is SpeechRecognition, which was used in the ArtGuide project.[14]

The SpeechRecognition library goes about as a wrapper for a few well-known discourse APIs and is accordingly incredibly adaptable. One of these is Google WebSpeechAPI. It constructs a standard API key that is complicated coded into the Speech Recognition library.

Speech to content interpretation is finished with the assistance of Google Speech Recognition. This requires a functioning web association with work. Because of the way that message bot is consistently up on the server, Google Speech Recognition is one of the least demanding and best to utilize.

- The API perceives in excess of 120 dialects and variations to help worldwide client base.
- It can return recognized text from audio stored in a file, as well as analyze short-structure and long-structure audio.

```
def voice_process(language_used):
    sound = AudioSegment.from_file(query_voice)
    sound.export(new_voice, format="wav")
    r = sr.Recognizer()
    with sr.AudioFile(new_voice) as source:
        audio = r.record(source)
    try:
        s = r.recognize_google(audio,
                                language = language_used)
        print(s)
    except Exception as e:
        s = e
    return s
```

Listing 2.5: The code implementation of SpeechRecognition

The principal functionality of SpeechRecognition occurs with the Recognizer class.

The text retrieved from the user's request then processes on the server-side. The search by unique code and name is currently available in the Telegram bot.

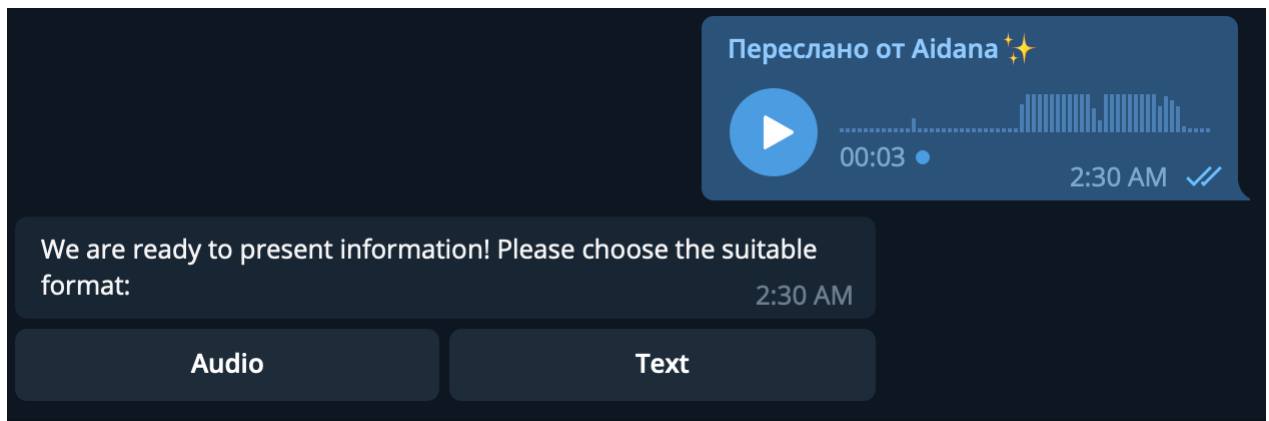


Figure 2.8: Audio chat with bot

2.2.6 Corresponding functions for commands

Bot commands give a more flexible way of communication between the client and the bot. It uses the `/command` syntax. A command must begin with the `'/'` sign and must not exceed 32 characters. Commands may use the names, numbers, and underscores in Latin. The ArtGuide has many commands below for using bot:

- `/start` - start the bot
- `/help` - returns a help message.
- `/language` - change the language of guide
- `/camera` - get information of gallery by taking the photo
- `/imagename` - get information of gallery by writing the name of gallery
- `/uniquecode` - get information of art by writing the unique code of the image

The Handler class is used to handle Telegram commands. The commands passed to the Bot API and handler add a list to the CallbackContext named `CallbackContext.args`. It may include a list of strings, which is the text on single or sequential whitespace characters following the split instruction. A message handler is a feature that is embellished with a `TeleBot` instance message handler decorator. Message handlers are made of one or more filters. For a certain request, each filter returns `True`, such that a request handler is qualified to manage the reply.[15]

As an example, the `get_help_info` function handles all the text messages that contain `help`, `imagename`, `camera`, and `unique code` commands. The bot sends commands to the function `command_text(text)` and to the specific client using `user_id` variable.

The `command_text(text)` is a general function, which is identified in which languages bot give a guide for a client. By default bot uses English.

```
@bot.message_handler(commands=['help',
```

```

'imagename', 'camera', 'uniquecode'])
def get_help_info(message):
    user_id = message.from_user.id
    text = message.text.replace('/', '')
    bot.send_message(user_id, command_text(text))

```

Listing 2.6: Function that handle commands

The Telegram Bot API library has a `@bot.callback_query_handler` decorator that passes a `CallbackQuery` object to a nested function. The callback is called when `check_update` has regulated that an amend should be altered by this handler. For convenience for the client, it is added buttons that are integrated directly into the message it belongs to. Inline buttons are accessible for both in chat mode and inline-mode messages received. Unlike custom response buttons, clicking inline keyboard buttons will not result in messages being submitted to the chat. Instead, buttons that function behind the scenes are enabled by inline keyboards: callback keys, URL buttons, and switch to inline buttons.

ArtGuide bot uses in-line keyboards when the bot asks the client in which format data should be presented in audio or text format. For this feature, `gen_markup()` function is created, where it returns a button, which is pressed by the client for further actions.

```

def gen_markup():
    keyboard = types.InlineKeyboardMarkup()
    audio_button = types.InlineKeyboardButton(
        text="Audio", callback_data="audio")
    text_button = types.InlineKeyboardButton(
        text="Text", callback_data="text")
    keyboard.row(audio_button, text_button)

    return keyboard

```

Listing 2.7: InlineKeyBoard buttons

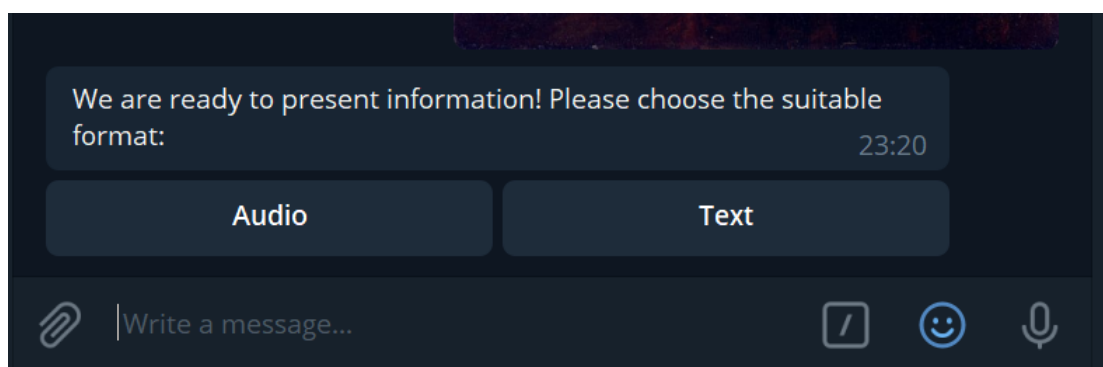


Figure 2.9: Inline keyboards in bot

For setup language, a guide is represented by callback buttons. If a user hits a callback icon, no notifications will be sent to the chat. Instead, the bot only gets the matching message. After receiving the message, the bot can show any result in a warning or a notification at the top of the chat screen. The function's language command used the callback buttons for switching the language of the guide.

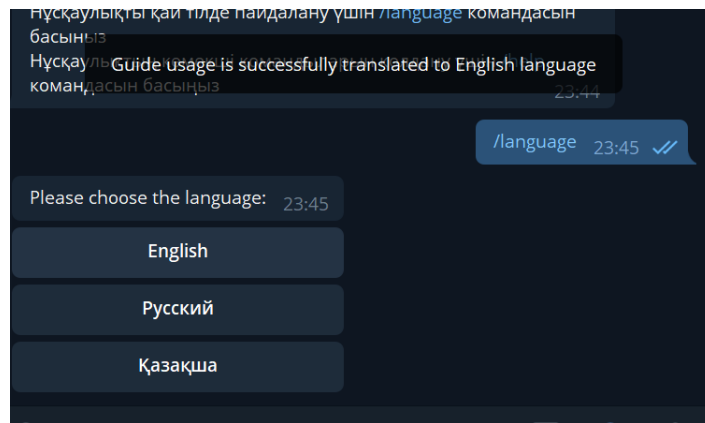


Figure 2.10: Callback buttons

Features related to search information by name and unique code of image are processed the same. At the beginning it is determined in which language guide is used through the function `command_text(text)`. Further using the filter query it finds data by image name and server return appropriate data in language which guide is set up. The same process is worked by searching unique code of a gallery. This piece of code shows how to retrieve data using filter query in the database.

```
for (b, c, d) in zip(image_names_eng,
image_names_kz, image_names_ru):
    x = b.replace("_", "").lower()
    y = c.replace("_", "").lower()
    z = d.replace("_", "").lower()

    if((x in res) or (y in res) or (z in res)):
        result = ArtGallery.objects.get(
            Q(image_title_eng__contains = b) |
            Q(image_title_kz__contains = c) |
            Q(image_title_ru__contains = d))
        text = command_text("info")
        bot.send_message(user_id, text,
            reply_markup = gen_markup())
```

Listing 2.8: Retrieving data from database using filter query

2.3 Implementing iOS and Android application

In the existing mobile industry, there are a number of smartphones with heterogeneous platforms like Android, iOS, Symbian, Windows Screen, BlackBerry, and others. Competition among the different operating systems is the key factor that triggers developers to introduce new functionality to the operating system as well as build viability. Statistics show that Android and iOS have taken the lead in the new smartphone industry. The journey began with creating a UI/UX template on Figma.[see AppendixA]

In developing mobile applications UX configuration is pivotal. By making a utilitarian model preceding coding will eliminate the measure of time spent being developed, notwithstanding getting plan defects early. Prototyping will assist coders with seeing precisely what to look for on the grounds by having a visual portrayal of how to imagine finished products.

In order to make a suitable plan were followed key factors:

1. The UI ought to be significant, clear, and reliable which would be unmistakable to clients.
2. Put related things together and separate inconsequential things, separate disparate things, and cause comparable things to take after each other.
3. The structure ought to be adaptable and lenient, decreasing the expense of errors and abuse by permitting fixing and re-trying.
4. The structure should make basic, normal errands simple, conveying unmistakably and essentially in the client's own language, and giving great easy routes that are seriously identified with longer methodology.

By having a landing page that extensively features the application thought the way toward making a storyboard was started. Checked how to implement primary highlights of the application, by observing their activities and adjusting prepared UI/UX to development.

The key feature of the project is image recognition of art objects in a gallery, so the “Art Guide” mobile application on ios and android platforms is based on usage of libraries Arkit and Arcore. The deep analysis of development is described in the next sections.

2.3.1 Mobile tools

There are differences between Android and iOS when it comes to the tools which serve as an environment for back-end application development. An Android is an open source system and Android SDK is available on well-documented API

heterogeneous platforms. The Android Studio tool is built to build a broad variety of mobile apps, and for the development language is used Java.

iOS is the platform for Apple's iPhone, iPad and iPod devices. Apple's primary language for iOS production is Objective-C but Swift is the most popular. Apple prohibits general context operations by approving its SDK, which ensures that an application's development cycle is even easier than Android. There are fewer tools for creating iOS based applications but the most popular one is Xcode 8. This IDE facilitates user interface (UI) prototyping without complicated scripting, which helps developers to build apps for all Apple products.

2.3.2 Database

iOS.

The role of the database in ArtGuide mobile app is to use the JSON file. JSON(JavaScript Object Notation)file is the most frequently used format for submitting and receiving data from Web services. The data exists in the form of pairs of key values. Using Swift Dictionaries it is simply to get values from the keys. Through converting the Data object, the JSONSerialization class is used to transform a JSON data into a key-value pair dictionary. The JSONSerialization class method JSON object(with options:) returns a form Any value and throws an error unless the data can be parsed. The mobile app has the same dataset as a telegram bot. Using the JSON file, it retrieved information by key-value principle.

Android.

There are some differences in the implementation of mobile app ios and android platforms. For saving information, it is used the arcoring tool database which is available in the ARCore SDK for Android. Arcoring is a command-line interface that creates an image database file and takes a collection of reference images

Firstly, it is created a snippet and add it to activity. Then it is created a class for an app for an augmented image. After it is disabled plane detection. This eliminates the handicon from the screen, which occurs directly after the initialization of the fragment and asks the user to move his mobile to look for the plane. It no longer wants this, because it does not consider a random plane but allows a particular gallery picture to be identified.

Then, the LATEST_CAMERA_IMAGE session update mode is mounted. It means that any time the camera frame is changed it can find out about image changes. A database is created as soon as the fragment is made. The code below shows how to handle frame fragment:

```
private void handleFrame(Fragment fragment) {  
    FragmentTransaction fragmentTransaction =  
        getSupportFragmentManager().beginTransaction();
```

```

        fragmentTransaction.setCustomAnimations(
            android.R.anim.fade_in, android.R.anim.fade_out);

        fragmentTransaction.replace(R.id.frame1, fragment);
        fragmentTransaction.addToBackStack(null);
        fragmentTransaction.commit();
    }

```

Listing 2.9: Frame fragment

2.3.3 Image recognition

With the arrival of the new Augmented Reality (AR) outline works "ARKit" for iOS and "ARCore" for Android in 2017, better approaches for communication with the cell phone have emerged. Rather than showing content level on the screen, a three-dimensional substance would now be able to be set in the client's immediate condition.[16]

Expanded truth is a field that is getting expanded subsidizing each year, as more organizations are understanding the capability of rendering virtual articles in reality. As the hardware gets increasingly marketed, the costs will get brought down while execution additionally goes up. Starting at now,an increased reality for the most part utilizes plane discovery and marker identification to discover and find objects. So as to join AI using deep neural networks to be able to recognize art objects in an augmented reality scene.[17]

iOS.

Augmented reality (AR) portrays client encounters that add 2D or 3D components to the live view from a gadget's camera such that it causes those components to seem to occupy this present reality. ARKit joins gadget movement following, camera scene catch, propelled scene preparation, and show accommodations to streamline the assignment of building an AR experience. You can make numerous sorts of AR encounters with these innovations utilizing the front or back camera of an iOS gadget.

The mobile development project is based on building up an iOS application, which consolidates utilization use of object detection, object acknowledgment. The main goal is to likely recognize if the mix of these fields is both plausible and alluring with present-day apparatuses accessible. The application should fill in as an option in contrast to traditional get-together manuals for art gallery objects.

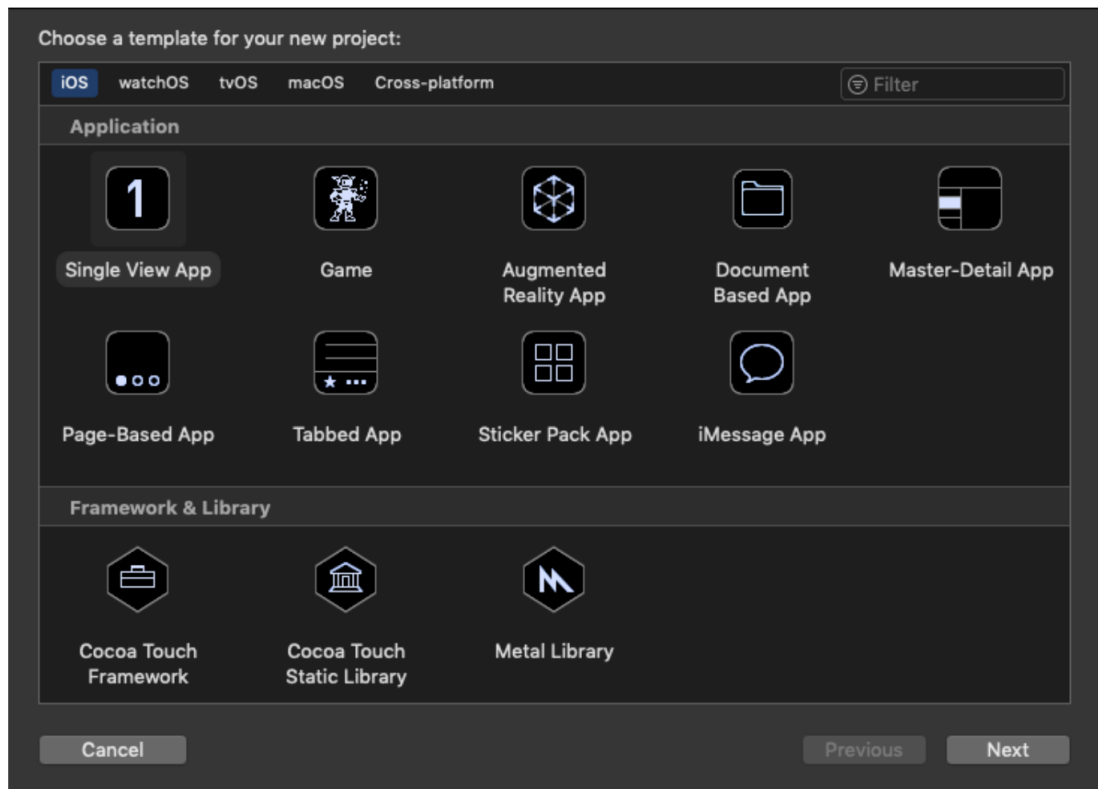


Figure 2.11: ARKit sample in Xcode

Apple’s equipment utilizes “core motion” computer vision algorithms which encourage gadgets to survey relative situations through the utilization of the camera. This is ARKit’s capacity to comprehend a situation and decipher a client’s relative area in it. ARKit utilizes custom code to interpret camera information and information from gadget sensors. The ARKit utilizes a strategy called visual-inertial odometry which is the mix of the data converged from iOS gadgets movement detecting equipment with a visual examination of the scene obvious from the telephone’s camera. The world following likewise dissects and comprehends the substance of the scene. Utilizing the hit testing strategy it can distinguish planes flat or vertical in the camera picture and tracks its position and size.[18]

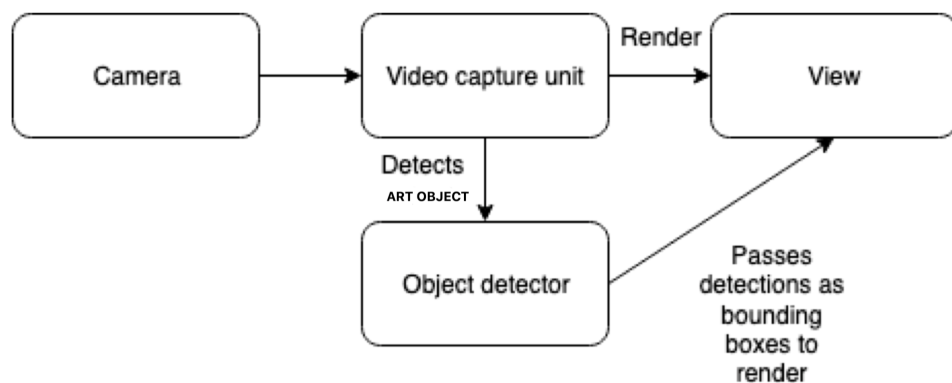


Figure 2.12: A logical overview of the object detection system.

An ARScene is contained inside an ARSCNView (AR Scene View) which additionally has an ARSession that deals with the movement following and camera picture preparing. For an ARScene to work, it must have a running ARSession. Planning a neural system for the picture classification is not the most effortless assignment and much work comprises attempting things and making astute suppositions. For classifying images in ArtGuide project Vision and Core ML frameworks were used. With the Core ML system, you can utilize a prepared AI model to group input information. The Vision system works with Core ML to apply order models to pictures, and to preprocess those pictures to make AI undertakings simpler and progressively dependable.

An ML model procedures input pictures in a fixed angle proportion, however, input pictures may have self-assertive perspective proportions, so Vision must scale or yield the picture to fit.

```
let model = try VNCoreMLModel(for: MobileNet()).model)
let request = VNCoreMLRequest(model:model,
completionHandler: {[weak self] request, error in
self?.processClassifications(for: request, error: error
)})
request.imageCropAndScaleOption=.centerCrop
return request
```

Listing 2.10: An ML model processes code implementation

Most models are prepared on pictures that are as of now arranged effectively for show. To guarantee the appropriate treatment of info pictures with subjective directions, pass the picture's direction to the picture demand handler.

Android.

The android platform can also provide users to create their own augmented world using their mobile phones, which is called the ARCore Development platform. With the assistance of a 2D picture following Android applications could perceive something like a page in a book, a film banner, or an artistic creation on the divider. Engineers could without much of a stretch cause their applications to acquaint objects with nature once the gadget perceived those 2D pictures.

- Programming interface for simultaneously (at a similar casing) obtaining the picture from a camera outline (without physically duplicating the picture cradle from the GPU to the CPU);
- Programming interface for getting shading amendment data for the pictures caught by the camera.

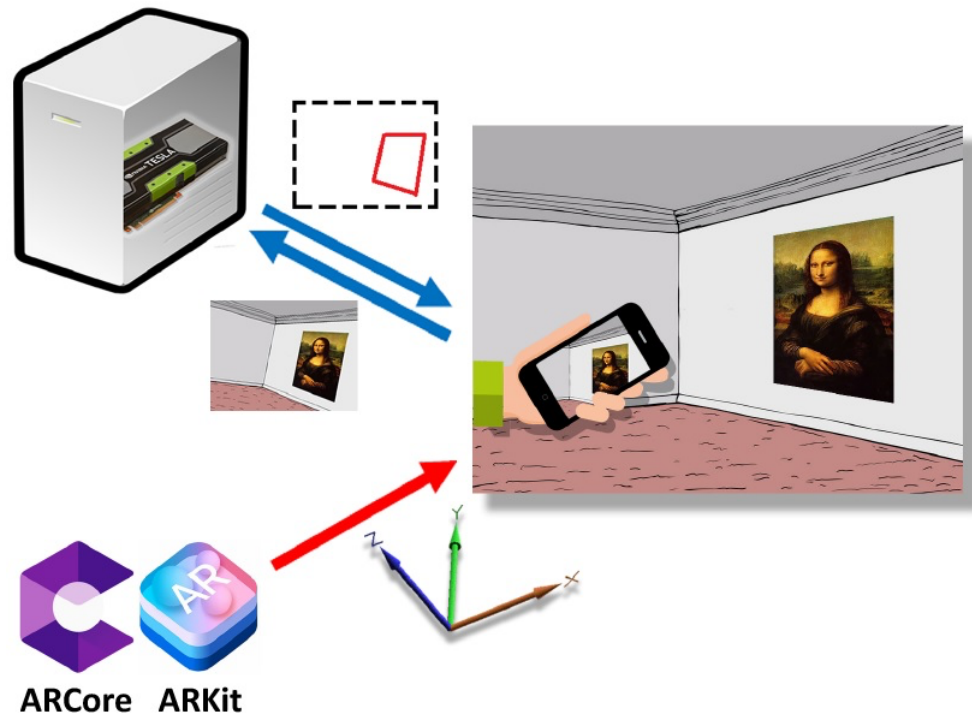


Figure 2.13: mobile Augmented Reality (AR) system

Further, ARCore utilizes a procedure called visual-inertial odometry, which includes distinguishing key physical highlights in the earth around the gadget. Those highlights are recorded from various edges as the gadget is moved and reoriented in physical space (moving is required; revolution doesn't give enough data). The pictures caught in this procedure are utilized together to get profundity; how people see profundity from two eyes.

It can react to and track pictures that are fixed set up, for example, a print holding tight a divider or a magazine on a table. As ARKit it starts following a picture, it gives assessments to position, direction, and physical size. These appraisals are ceaselessly refined as ARCore assembles more information. Giving an exact physical size improves picture location time. On the off chance that no physical size is determined, ARCore gauges the size.

ARCore's movement following innovation utilizes the telephone's camera to distinguish intriguing focuses, called highlights, and track show those focuses move after some time. With a mix of the development of these focuses and readings from the telephone's inertial sensors, ARCore decides both the position and direction of the telephone as it travels through space.

So as to assemble the preparation dataset for the system, another methodology is utilized. Rather than marking every individual preparing picture with respect to the standard administered learning, the underlying directions of the patches are proliferated on the various pictures utilizing the posture of the camera for all the photos. This is conceivable with the help of AR-Core designed by Google.

2.3.4 Functions for playing audio

After processing the gallery using image recognition, the app should return information in text or audio format. Sending audio format is implemented using libraries of platforms iOS and Android.

iOS.

In iOS, playback of audio files is implemented using the AVPlayer class that provides the mechanism to monitor the transportation actions of the player. AVPlayer class originated from AVFoundation. AVFoundation is one of the tools that can be used to recreate and build audiovisual material dependent on time. This offers an Objective-C interface that utilizes the time-based audiovisual data to function on a comprehensive basis.[19] It will use this to analyze, build, modify, or re-encode media files as an illustration.

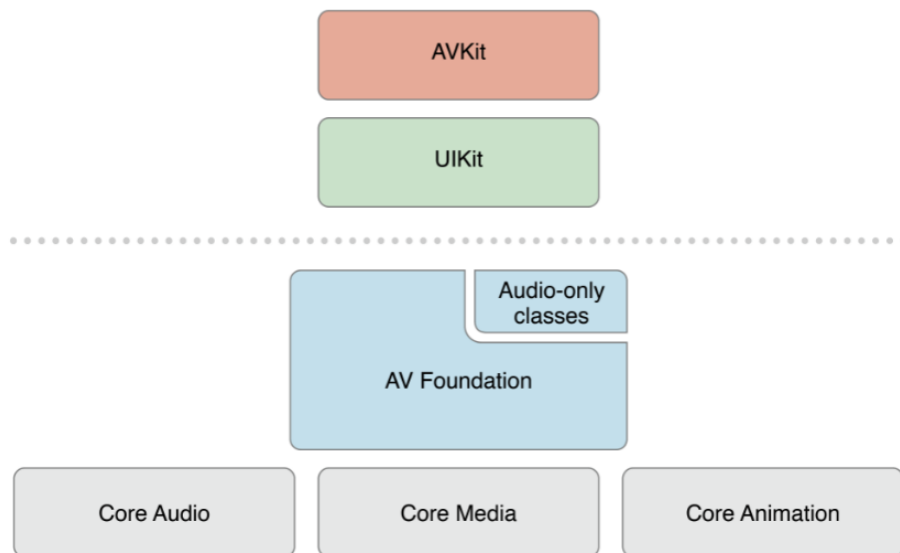


Figure 2.14: AVFoundation stack on iOS

The primary class used for media coverage by the AV Foundation system is AVAsset. An AVAsset instance is an aggregated description of one or more pieces of media data (audio and video tracks) gathered. It includes details on the set as a whole, such as its theme, length, the normal scale of the display, etc. Each of the media data pieces in the asset is of a uniform kind and is called a track. Assets may include metadata, as well. In the AV Foundation, a crucial principle is that initializing an element or a track does not automatically imply that it is ready for usage. Sometimes the length of an object (for example, an MP3 file can not include descriptive information) may take some time to measure.

An AVPlayer object is used to manage the playback of the content. A player is a controller that is used to monitor an asset's playback, such as beginning and stopping playback and looking for a certain date. For ArtGuide single audio information is played using an AVPlayer instance.

When playing an element it does not explicitly supply assets to an AVPlayer entity. Instead, it does include an AVPlayerItem instance. A player object controls the condition of the appearance of an element to which it is connected. A player object includes player element tracks — AVPlayerItemTrack instances — which correspond to the asset tracks.[20]

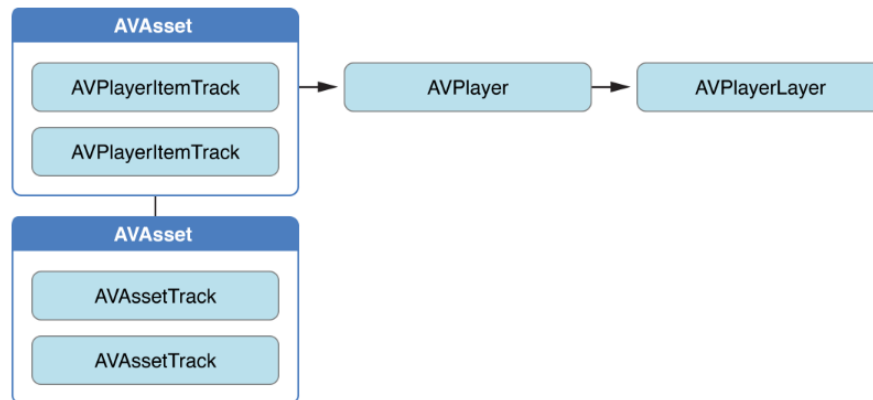


Figure 2.15: Playing assets

Android.

In the Android platform, playback of audio files developed using the MediaPlayer class. Audio/video files and streams can be playback managed with MediaPlayer class. As MediaPlayer class is thread-safe. Production of and all links to instances of players should be on the same thread. If callbacks are registered the thread must have a Looper.

Playback control is conducted as a state machine for audio/video files and streams. Figure 2.16 indicates the life cycle and status of an object controlled on a MediaPlayer by the assisted playback control operations. The ovals reflect the states in which an object of the MediaPlayer can reside. The arcs reflect the operations of playback control which drive the transition from the object state. There are two Arc forms. The arcs with a single arrowhead are synchronous method calls, whereas asynchronous method calls are those with a double arrow head.[21]

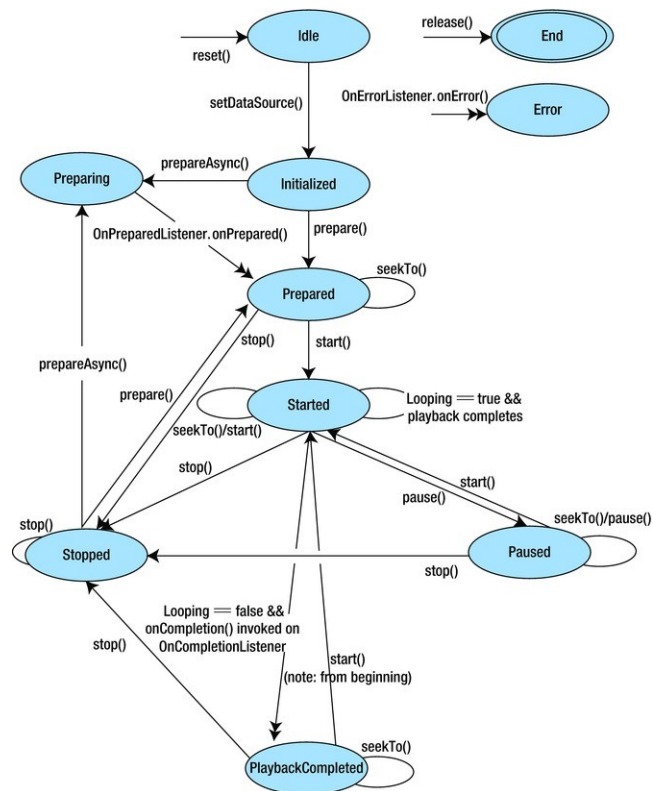


Figure 2.16: State diagram of MediaPlayer

If a MediaPlayer object is created using a new or after reset() method, which means the inactive zone; and after calling function release(), it is in the Final status. The MediaPlayer object's life-cycle is between these two conditions.

```

MediaPlayer = new MediaPlayer();
try {
    mediaPlayer.setDataSource(
        afd.getFileDescriptor(), afd.getStartOffset(),
        afd.getLength());
    mediaPlayer.prepare();
} catch (IOException e) {
    e.printStackTrace();
}
mediaPlayer.start();
  
```

Listing 2.11: Code for MediaPlayer

2.3.5 Additional features

Floating panel (for design)

Floating Panel is a straightforward and simple to utilize UI segment for another interface presented in Apple Maps, Shortcuts, and Stocks application. The new

interface shows the related substance and utilities in equal as a client needs.
Key features:

- Simple compartment see controller;
- Fluid activity and signal dealing with;
- Scroll see following;
- Common UI components: Grabber handle, Backdrop and Surface adjusting corners;
- 2 or 3 grapple positions(full, half, tip);
- Layout customization for all quality environments(i.e. Scene direction support);
- Behavior customization;
- Free from basic issues of Auto Layout and signal taking care of.

```
let storyboard = UIStoryboard(name: "Main", bundle: nil);
let vc=storyboard.instantiateViewController(withIdentifier:
"pictureInfoText") as! TextInfoViewController
let options = [ SemiModalOption.pushParentBack: true ]
vc.view.frame = CGRect(
x: 0, y: 0, width: UIScreen.main.bounds.width, height: 350)
vc.view.backgroundColor = UIColor.white
presentSemiViewController(vc,options:options,completion:{
    print("Completed!")
}, dismissBlock: {
})
```

Listing 2.12: Code implementation of floating panel (Swift)

```
public class BottomSheetDialogFragment
extends AppCompatActivity
```

Listing 2.13: Code implementation of floating panel in Android

In Android the analog of floating panel is **BottomSheetDialogFragment**, which is called **BottomSheetDialogFragment**, which extends from **AppCompatActivity**.

It is completely gesture-driven, considers safe territory insets, has support for option to-left dialects heated in and consequently responds to the on-screen console. Contrasted with numerous other open-source board arrangements, floating panel is intended to be a constantly obvious youngster see controller, and thus

does not utilize the custom view controller change API.

Natural language programming

Since artificial neural networks enable nonlinear processes to be modeled, they have become a very common and useful method for solving many problems, such as speech recognition, computer vision, visualization, and others. A broad variety of expertise allows for the usage of artificial neural networks in Natural Language Processing (NLP) activities.

iOS.

One of my preferred software engineering electives was Speech Synthesis and Recognition since Natural Language Processing and Computational Linguistics are getting increasingly across the board.[22]

iOS has a worked in discourse interpretation framework, which permits you to change over any voice recording into text. The Speech system perceives verbally expressed words in recorded or live sound and afterward makes an interpretation of substance into the content. This system works without the nearness of the console. In the venture actualized discourse acknowledgment to perceive verbal orders or handle content correspondence in voice acknowledgment part of the application.[23]

```
import Speech
var recognitionRequest:
    SFSpeechAudioBufferRecognitionRequest?
var recognitionTask: SFSpeechRecognitionTask?
let audioEngine = AVAudioEngine()
var speechRecognizer = SFSpeechRecognizer(
    locale: Locale.init(identifier: "en-US"))
```

Listing 2.14: Code implementation of NLP (Swift)

Android.

Android SDK has the potential to incorporate consumer speech recognition tools effortlessly and turn voice into text. Apps like Google Keep, Any. DO allow relatively innovative use of speech-to-text conversion functionality. Android app can understand his / her speech as soon as a client says it, and translate it into text. It will do using the RecognizerIntent class. Using the voice to text service does not need any internet access. It runs in Offline mode.[24]

```
String language = "en-US";
Intent intent = new Intent(RecognizerIntent.
    ACTION_RECOGNIZE_SPEECH);
```

```
        intent.putExtra(RecognizerIntent.  
            EXTRA_LANGUAGE_MODEL, language);  
intent.putExtra(RecognizerIntent.  
    EXTRA_LANGUAGE, language);  
intent.putExtra(RecognizerIntent.  
    EXTRA_LANGUAGE_PREFERENCE, language);  
intent.putExtra(RecognizerIntent.  
    EXTRA_ONLY_RETURN_LANGUAGE_PREFERENCE, language);  
startActivityForResult(intent, REQUEST_CODE_SPEECH_INPUT);
```

Listing 2.15: Code implementation of NLP in Android

An Intent called `RecognizerIntent` in the above code which asks for input of speech and then sends it via speech recognizer. It does through `ACTION_RECOGNIZE_SPEECH`.

Chapter 3

Conclusion

Tour guides are increasingly becoming ancient history, as smartphone apps provide tourists with immersive tours of the most revered museums in the world. To help users better appreciate everything from ancient art to prehistoric bones, cell phone devices are increasingly becoming important to intrepid museum-goers. A smartphone device should be regarded as a great opportunity in the life of a wide variety of customers to offer cultural and creative material.

Museum knowledge guide production would draw more visitors to a deeper study of the past. Although the key function of the software is the visual search engine, which provides the customer with a perfect ability to navigate the gallery without guides.

To scan for significant information data through a visual pursuit is more noteworthy than the time it takes to process a text. The expansion in the number of articles/distractors will build the hour of an inquiry. For instance, a quest for a coordinating picture in a database of thousands of pictures takes quite a while; since the inquiry needs to experience the database and locate the best match. The image search engine involves an expectation to learn and adapt to the underlying stage. The search index experiences the underlying AI stage. For getting a more accurate result of the sent photo by a client, it is tested several image detection methodologies. The testing is based on some external factors which have an influence on an accurate result. These external factors include a bad-quality taken photo, unfocused image, the distance between client and figure, and so on. Taken into account those factors, a bot used SURF methodology for image regression rather than ORB and SIFT methods of search engines. Because SIFT methodology performs quicker results than SIFT without lessening the nature of the recognized focuses. By cropping the size of the sent image, it helps to find more accurate points during image regression. It is taken into account when the client takes a photo from close or further distances. The gotten result gives information in two formats where users can read or listen to audio in the chosen language.

3.1 Future development

With the daily development of the mobile industry, mobile tools need more new features to simplify user usage as much as possible. For this reason, the implementation of ArtGuide and the bot need several potential technologies upgrades. The major upgrade to the database should have been made. Because these three programs use separate resources, it would be easier to access details from a single database server. As an illustration, MySQL server could be used. Given the pressure on the cloud, it needs different ideas to boost the floating on the server.

To produce better search results, it would have required the image filtering method. In this method, if the user uploads a blurred or inappropriate picture and it's difficult to get function or key-points in a file, then the server will ask the user for a better image. Filtering effects may also be achieved by choosing the picture with the highest corresponding local features or key-points with the input picture, and taking that image as the output. It only shows material specific to the type of images.

As the ArtGuide is being introduced into the individual museum, it would be easier to include the museum's interactive map. This will make navigation in the museum easier, particularly to those who first came to the museum. Additionally, locations that were explored around the museum should have been highlighted, rendering this feature a helpful resource.

The last improvement is adding a virtual tour to the museum. It will give guests a virtual sense of being inside the museum and looking at the art in the individual galleries. As an example, The Hermitage Museum and Gardens in St. Petersburg uses this feature in their own mobile tools. Additionally, an online interactive tour of the museum of web edition would have been anticipated. According to many museums around the world such as the LOUVRE museum, which provides free online tours of some of its most significant and famous exhibitions such as its Egyptian Antiquities and Michelangelo works. It would be possible to implement a 360-degree look at one of the museums in Kazakhstan and click on the unique objects to get more detail regarding their past.

Appendix A

Appendix

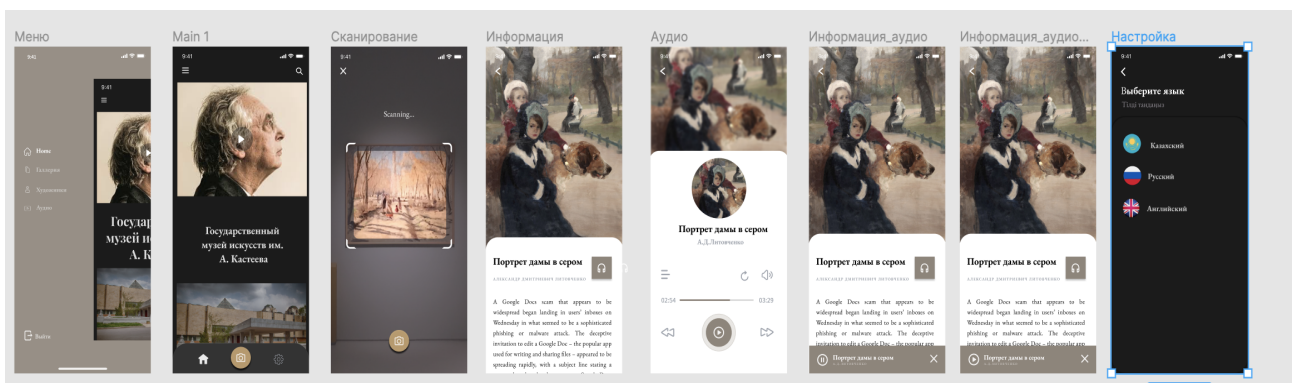


Figure A.1: Design for iOS and Android

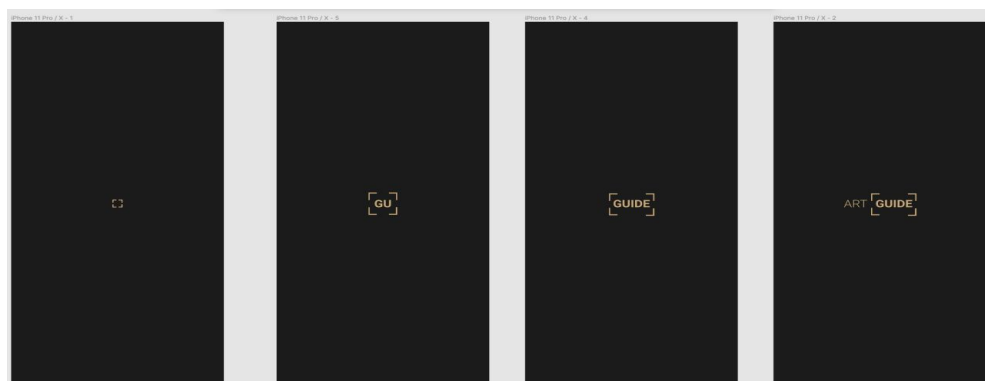


Figure A.2: Animation and logo for iOS and Android applications

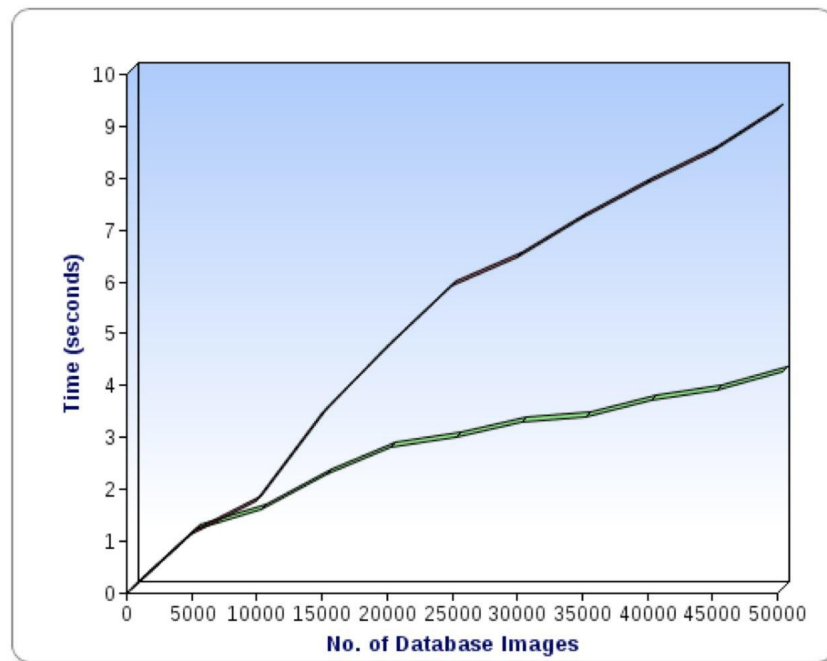


Figure A.3: Search time of dataset Images

```
[
  {
    "uniquecode": "44556",
    "imagenname_kk": "Мона Лиза",
    "imagenname_en": "Mona Lisa",
    "imagenname_ru": "Мона Лиза",
    "text_kk": "«Мона Лиза» - әйгілі кескіндеме туындыларының бірі Леонардо да Винчидің суреті. Жазудың нақты күні белгісіз (кейбір",
    "text_en": "Mona Lisa - a painting by Leonardo da Vinci, one of the most famous works of painting. The exact date of writing is",
    "text_ru": "«Мона Лиза» - картина Леонардо да Винчи, одно из самых известных произведений живописи. Точная дата написания неизв",
    "img_name": "Мона Лиза"
  },
]
```

Figure A.4: Parsing JSON data

References

- [1] Harsh Patel. “VISUAL SEARCH APPLICATION FOR ANDROID”. In: (2012). DOI: <https://doi.org/10.31979/etd.zf83-mgzy>.
- [2] Navi Mumbai. “Use of Agile Methodology for Mobile Applications”. In: *International Journal of Latest Technology in Engineering, Management and Applied Science (IJLTEMAS)* (October,2016), p. 77. DOI: <https://ssrn.com/abstract=2298361>.
- [3] Dr. Swati V. Chande² Harleen K. Flora¹. “A REVIEW AND ANALYSIS ON MOBILE APPLICATION DEVELOPMENT PROCESSES USING AGILE METHODOLOGIES”. In: *International Journal of Research in Computer Science* 4 (2013), pp. 9–18.
- [4] Gianpaolo Basile Federica Palumbo Gandolfo Dominici. “Designing a mobile app for museums according to the drivers of visitor satisfaction”. In: (July 25, 2013), p. 8. DOI: <https://ssrn.com/abstract=2298361>.
- [5] Avinash Sharma. *Reasons why we Trust Agile for our Mobile App Development Process*. <https://appinventiv.com/blog/>. 2018.
- [6] Arez Khalimi. *How to Create and Deploy a Telegram Bot using Python?* <https://djangostars.com/blog/>.
- [7] *Marvin’s Marvellous Guide to All Things Webhooks*. <https://core.telegram.org/bots/webhooks>.
- [8] Django Documentation. *Migrations*. <https://docs.djangoproject.com/en/3.0/topics>.
- [9] M Venu Gopala Rao M Manno M. Neelima. “Image classification using Deep learning”. In: *International Journal of Engineering and Technology* (2018), pp. 614–618. DOI: 10.14419/ijet.v7i2.7.10892.
- [10] Tinne Tuytelaars Herbert Bay and Luc Van Gool. “Speeded-up robust features (SURF) [Computer vision and image understanding]”. In: 110.3 (2008), pp. 346–359.
- [11] David G Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: 50.2 (2014), pp. 91–110.

- [12] Hafize Keser Burcu Turan. “Museum Guide Mobile App: The Case of the Near East University Classical Car Museum”. In: 131 (May,2014), pp. 278–285. DOI: <https://doi.org/10.1016/j.sbspro.2014.04.117>.
- [13] Hong Cui Anne E. Thessen and Dmitry Mozzherin. “Applications of Natural Language Processing in Biodiversity Science”. In: (2012), p. 17. DOI: 10.1155/2012/391574.
- [14] Hong Cui Anne E. Thessen and Dmitry Mozzherin. “REVIEW ON NATURAL LANGUAGE PROCESSING”. In: (2013).
- [15] pyTelegramBotAPI documentation. *Python implementation for the Telegram Bot API*. <https://pypi.org/project/pyTelegramBotAPI/>.
- [16] Raffaele Boccaccini Antonio Laudazi. “Augmented Museums Through Mobile Apps”. In: (2014), p. 17.
- [17] Ondrej Popelka David Prochazka Michael Stencl and Jiri Stastny Mendel. “MOBILE AUGMENTED REALITY APPLICATIONS”. In: (2011).
- [18] Iván Ruiz-Rube José Miguel Mota Macias. “Augmented reality mobile app development for all”. In: (September,2017). DOI: 10.1016/j.compeleceng.2017.08.025.
- [19] Apple documentation. *AVFoundation Programming Guide*. <https://developer.apple.com/library/archive/documentation/>.
- [20] Apple documentation. *AVPlayer*. <https://developer.apple.com/documentation/avfoundation/avplayer>.
- [21] Android documentation. *MediaPlayer*. <https://developer.android.com/reference/android/media/>.
- [22] Martin Mitrevski. “Natural Language Processing on iOS”. In: Jan. 2018, pp. 165–185. ISBN: 978-1-4842-3395-5. DOI: 10.1007/978-1-4842-3396-2_6.
- [23] MOHD SANAD ZAKI RIZVI. *Create Natural Language Processing based Apps for iOS in Minutes! (using Apple’s Core ML 3)*. <https://www.analyticsvidhya.com/blog>. 2019.
- [24] SHANGLUN WANG. *How to Build a Natural Language Processing App*. <https://www.toptal.com/algorithms/>. 2019.

List of Figures

2.1	Principles of Agile methodology	10
2.2	PythonAnywhere interface	12
2.3	illustration of mechanisms Long Polling and Webhook	13
2.4	ERD of bot's table	15
2.5	The matching of varying intensity images using (a) SIFT (b) SURF (c) ORB.	17
2.6	Comparison between algorithms.	18
2.7	Speech recognition block diagram	19
2.8	Audio chat with bot	21
2.9	Inline keyboards in bot	22
2.10	Callback buttons	23
2.11	ARKit sample in Xcode	27
2.12	A logical overview of the object detection system.	27
2.13	mobile Augmented Reality (AR) system	29
2.14	AVFoundation stack on iOS	30
2.15	Playing assets	31
2.16	State diagram of MediaPlayer	32
A.1	Design for iOS and Android	38
A.2	Animation and logo for iOS and Android applications	38
A.3	Search time of dataset Images	39
A.4	Parsing JSON data	39

Listings

2.1	Setting Webhook	13
2.2	TelegramBotView class	14
2.3	The code implementation of SURF algorithm	18
2.4	The implementation of processing query image and dataset	19
2.5	The code implementation of SpeechRecognition	20
2.6	Function that handle commands	21
2.7	InlineKeyBoard buttons	22
2.8	Retrieving data from database using filter query	23
2.9	Frame fragment	25
2.10	An ML model processes code implementation	28
2.11	Code for MediaPlayer	32
2.12	Code implementation of floating panel (Swift)	33
2.13	Code implementation of floating panel in Android	33
2.14	Code implementation of NLP (Swift)	34
2.15	Code implementation of NLP in Android	34