

SOFTWARE ANALYSIS AND DESIGN



<QUESTION ANSWERING PROJECT>

İkram YURTDAS 05150000619

Erdem ÇALIŞIR 05150000750

Mine AYGÜN 05150000704

VISION

INTRODUCTION

There are many music listening platform on the market such as Spotify, iTunes, Fizzy etc. There are many music and artists on these platforms. These are constantly growing and updating by making agreements with other companies. Songs of newly released singers can be upload to many platforms with different qualities, however Spotify is a corporate company and they have organized the music and their quality. Therefore Spotify the largest of these platforms.

Spotify restricts this to copyright only Spotify has a detailed search menu for songs but doesn't have much information about singers. However this information is only superficial. Our project which we make, aims to improve and help to Spotify hold singers information. Using this information together we can provide a more useful platform every listener.

BUSINESS OPPORTUNITY

This program will be used by two types of users according to our opinion. Actors: Listener Producer Companies Spotify attracts a lot of users. This layout needs innovations to be continuous. For this reason people can be used our information software. Listener can be reach more information about singers clearly and quickly. If there is movement, there must be music. Therefore companies are constantly seeking music. Companies should be able to reach the singer as well. These singer information are scattered. Our program collect together these information. Accompanied by this information, companies offer to listeners. These offers can be used commercial purpose. Such as movie, advertising, tv series etc. Our program helps to companies choices with their specification.

KEY HIGH LEVEL GOAL AND PROBLEMS OF THE STAKEHOLDER

High-Level Goal	Priority	Problems and Concerns	Current Solutions
mixed data incorrect data	high	presence of incorrect information	retrieving data randomly from the internet
user friendly interface	low	can be confused for user	coloring and use of specific interfaces
insufficient information	middle	information the user is looking for may not be showing	shows what's in the resources
...

SUMMARY OF THE BENEFITS

Supporting Feature	Stakeholder Benefit
Functionally, when the information about the artist is searched, the correct answer will be shown.	Automated, fast point-of-search artist services.
detection of missing data and their completion	ease of filling from the outside
addition of answers to the problem that was not previously in the system	high scope of information
....

ASSUMPTIONS AND DEPENDENCIES

- Spotify provides information parts related to music.
- Wikipedia provides information parts related to singer personal data.
- They are available on Internet for free and we expect it to continue this way.

THE PROCEDURES OF USE CASE

Step 1: Choose the System Boundary

For this case study, the API system itself is the system under software design; everything outside of it is outside the system boundary.

Steps 2 and 3: Find Primary Actors and Goal

Actor:

Goal:

Listener

Get information

Interest singers

Get data for their curiosity

Producer

Company get information

Satisfy the user's demands Step

Step 4: Define Use Cases

In general, define one use case for each listener goal. Name the use case similar to the user goal for example, **Goal:** get information; **Use Case:** Ask questions about artists

PRECONDITIONS

UC1: Ask questions about artists

- Listener and company should not ask about the content of the albums. The program is not yet capable of turning album information
- The program cannot yet provide information about songs
- Questions should be about artists and it should be able to ask the following questions
 - internet address of artist's information
 - the popularity of the artist
 - the number of followers that artist has
 - song types that artist sings
 - where the artist was born
 - age and date of birth of the artist
 - album's names of the artist, the number of albums in the database
 - The number of album should be maximum 50
 - Album's names do not support the Turkish characters
- The popularity order in Turkey
- older and younger artist

UC2: Entering the question patterns from the interface.

- The user must fill in the necessary fields of the question type. If it is not filled, its input will be rejected and warned by red color.
- The required amount of free space should be available where the data is saved.

UC3: Get data from Internet

- The user must fill in the necessary fields of the question type. If it is not filled, its input will be rejected and warned by red color.

FULLY DRESSED STYLE

Use case UC1: Ask questions about artists

Scope: Information application with API

Level: User goal

Primary Actor: Listener

Stakeholders and Interests:

Listener:

Wants to learn information, to obtain accurate information, to satisfy your curiosity and wants visible display and fast service with minimal effort

Producer Company:

Wants to learn information, to obtain accurate information, to satisfy listener's interests and wants fast service.

Success Guarantee (or Post conditions):

The information which the user is looking for is accurately shown on the screen.

Main Success Scenario (or Basic Flow):

1. The query panel is opened to the user.
2. The query panel shows the artists that are registered in the system.
3. User asks about artists. ,
4. The user's question is split to parts.

5. Parts are searched for matches.

6. Parts and data set are matched

7. The result is displayed

Extensions (or Alternative Flows):

2. The data of artists that are registered in the system may be missing.

a. Missing information is expected to be filled by user

b. The data set can be accurately updated by system

3. Use of non-English character

a. the error is displayed on the screen

i. User must be use English character

b. The data set can be updated and Turkish character support can be provided.

5-6. Matching cannot be found

a. The data set can be expanded

i. Users can fill data set

ii. The data set can be updated

b. The number of matching are 2 and more than 2

i. Matches are displayed as alternative sorted ways

7. The result is empty

a. User can add new registers

b. The system can be expanded

Special Requirements:

- Fast internet
- Fast PC
- User-friendly display
- Language internationalization on the text displayed

Technology and Data Variations List:

1. The query panel is read by voice assistants.
3. The question can be entered as a voice
7. The result can be read out loud

Frequency of Occurrence:

Could be nearly continuous.

Open Issues:

- Who is the singer*?
- Tell something about the singer*?
- How many followers does singer have*?
- What is the type of the singer*?
- When was born singer* ?

hint: *singer should come instead of “singer name”(such as Ceza)

Use case UC2: Entering the question patterns from the interface.

Scope: Entering information application

Level: User goal

Primary Actor: User

Stakeholders and Interests:

User:

Wants to enter question pattern from the interface Wants to see older data Wants to update data in dataset

Success Guarantee (or Post conditions):

Process successful. New question pattern has been added to the system or has been updated. The system can answer the questions.

Main Success Scenario (or Basic Flow):

1. The query panel is opened to the user.
2. The sample pattern that the user can enter is shown
3. User enters a question pattern
4. The question pattern is saved to storage
5. The system can answer this question.

Extensions (or Alternative Flows):

- 2a. The pattern in the sample may not be understood by the user.
 - a. Sample can be written more clearly
- 2b. There may not be data that is suitable for the pattern in the sample that the user enters.
 - a. Change blank fields with null fields
 - b. System can wish blank fields must be filled
3. The data entered is not suitable for the question pattern
 - a. Error occurred
 - i. Data entered should be corrected
 - b. Select appropriate data
4. Memory is full
 - a. Storage can be enlarged
 - b. The older data is deleted and new data is written

Special Requirements:

- Words can be completed automatically
- User-friendly display
- Language internationalization on the text displayed

Technology and Data Variations List:

1. User can enter data in written or verbal
2. Articles can be written in different colors
 - a. The required field can be written in red
3. User can enter data in written or verbal

Frequency of Occurrence:

Could be nearly continuous.

Open Issues:

When (artistName) was born?

Who is (artistName)?

How many followers does (artistName) have?

Use case UC3: Get data from Internet

Scope: API application

Level: Subfunction

Primary Actor: Software

Stakeholders and Interests:

Software:

Wants to get information, to obtain accurate information, and wants visible display and fast service with minimal effort

Success Guarantee (or Post conditions):

The system gets JSON data and saves data to storage.

Main Success Scenario (or Basic Flow):

1. Data which is searched is selected
2. Search in specific addresses on the Internet
3. Found the matching API file
4. Received the required data
5. Data is processed

Extensions (or Alternative Flows):

1. Out of topic data was requested
 - a. Error occurred
 - i. Requested data about the topic
2. Cannot reach internet addresses
 - a. Redirect to the alternative addresses
 - b. Search in local storage
3. File is bad which is got from API
 - a. Send a report to server
 - i. Required an update
 - b. API file is updated
 - i. Update process by user
 - c. Ignored bad files
- 4-5 . The older data is deleted and new data is written
 - a. Delete non necessary ones
 - b. Fix the extra data
 - i. Fixed by User
 - ii. Fixed by System

Special Requirements:

- Different api s can be added

Technology and Data Variations List:

4. Gson or Json file can be used

Frequency of Occurrence:

Could be nearly continuous.

Open Issues:

How can i get the data in different ways?

SUPPLEMENTARY SPECIFICATION

Introduction

Functionality

(Functionality common across many use cases)

Logging and Error Handling

Errors are not recorded somewhere but can be observed instant

Pluggable Rules

At various scenario points of several use cases (to be defined) support the ability to customize the functionality of the system with a set of arbitrary rules that execute at that point or event.

Security

All use, users thought to be anonymous

Usability

Human Factors

Read and write words incorrectly

Reliability

Recoverability

It is used locally after data is received. Even if data from outside is lost, we will have data. If the data we have is lost, we will supply it externally

Performance

As mentioned under human factors; users very quickly. But our system is no bottleneck. Our goal: answer in less than 10 second

Supportability

Adaptability

The system is currently working for artists. It can be easily adapted when used in different areas in the future. For example, footballer, politician ..

Configurability

Our system is independent of panels. This software will work in different compilers. There will be nothing changing as long as the necessary downloads are made. Much more analysis is needed in this area to discover the areas and degree of flexibility, and the effort to achieve it.

Implementation Constraints

ArtistSearchSystem leadership insists on a Java technologies solution, predicting this will improve long-term porting and supportability, in addition to ease of development.

Free Open Source Components

In general, we recommend maximizing the use of free Java technology open source components on this project. Also we use free api data and personal data

Legal Issues

We must act within the scope of protection of personal data. In the future, licensed data sets must be obtained.

Information in Domains of Interest

Pricing

As the application is the application of the software, the price will increase as the data set expands.

Application-Specific Domain (Business) Rules

ID	Rule	Changeability	Source
RULE 1	data cannot be copied when received with a license	low	copyright
RULE 2	If it is requested to be copied, it must pay the fee, such as credit card information must give the data.	medium	e-commerce contracts
...

DOMAIN MODELS

How to Create a Domain Model?

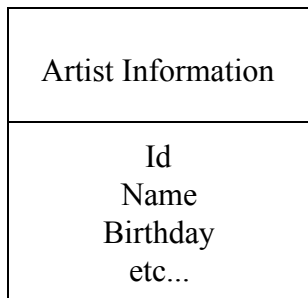
1. Find the conceptual classes (see a following guideline).

User	Question	Question Store
Answer	Question Analyzer	Artist Store
Web Service	Artist Information	

2. Draw them as classes in a UML class diagram.

3..Add associations and attributes.

Example attributes for Artist



Complete Domain Model

