Software Requirements Specification

for

ARTYFY

Prepared by

Team - 5

Advanced Software Engineering

Table of Contents

Table of Contents **2**
1. Introduction **3**
1.1 Purpose 3
1.2 Scope 3
1.3 Definitions, Acronyms and Abbreviations 3
1.4 Intended Audience and Reading Suggestions 3
1.5 References 3
1.6 Overview 4
2. General Description **4**
2.1 Product Perspective 4
2.2 Product Functions 4
2.3 User Classes and Characteristics 4
2.4 General Constraints 6
2.5 Assumptions and Dependencies 6
3. Specific Requirements **6**
4. External Interface Requirements **11**
4.1 User Interfaces 11
4.2 Hardware Interfaces 13
4.3 Software Interfaces 13
4.4 Communications Interfaces 13
5. Other Non-functional Requirements **13**
5.1 Performance Requirements 13
5.2 Safety Requirements 13
5.3 Security Requirements 14
5.4 Software Quality Attributes 14
6. Other Requirements **14**
6.1 Database Requirements 14

1. Introduction

1.1 Purpose

The purpose of our project is to develop a high-end, user friendly, cost effective Gallery and Art Management engine that helps intended target users to keep track of all the local galleries, Museums, Artists and Art exhibitions. The software bridges the existing artistic platforms (Arts Gallery's & Artists). It can create new marked space for the Art lovers. The Art finder application will use image and pattern recognition techniques to classify the paintings based on the Artist name or Art Style. This will help the students as well as the museum staffs to group the various paintings into the required category, by just uploading the image of the art. This image comparing it with the image details in the database.

1.2 Scope

The scope of this project is to develop an authentic android application that is used for Artists or Art gallery owners to showcase their arts and also sell their arts respectively. The application can also use for search nearby art galleries or museums. The project will also allow users to view or purchase all categories of arts available.

1.3 Definitions, Acronyms and Abbreviations

In this document, each feature has its own importance and plays a prominent role to develop based on the user necessities. the equal importance must provide for all the requirements and the features which are listed in the document. The definition of the terms used in the document are

Artyfy – refers the application name.
Owner – refers the Art gallery or museum owners.
Artist – refers the artist who wants to present their arts.
User – refers the end users who wants to explore the arts.

1.4 Intended Audience and Reading Suggestions

The intended audience for the document is product owners, product managers, developers, and testers. The product owners review the features and requirements which are available in the document. According to the document, the product managers list the requirements and prioritize the features. The developers and testers refer the document to understands the functionality of the features. The product owners and project managers must go through every detail which are present in the document and the whole document is recommended for both the developers and testers, but they can directly access the information which is relevant.

1.5 References

- https://www.eventbrite.com.au/blog/event-registration-form-template-ds00/
- https://wpamelia.com/login-page-design/

- https://www.thinkwithgoogle.com/intl/en-ca/marketing-strategies/app-and-mobile/near-mesearches/

1.6 Overview

In today's world, Due to ongoing pandemic it would be difficult for user to find arts in new places and also for artists to showcase or sell their arts. In order to overcome this problem a software is needed which lists nearby museums and their respective art galleries of different categories and also provides bridge between art sellers and buyers. The users use the artyfy application to explore the nearby art galleries based on their current location or preferred location. The users trust the application to provide user details due to its confidentiality and security. This application will help all the art lovers who want to know explore the arts which are presented in art galleries and museums. Our application exhibit nearby museums and list of total artists who are registered to showcase their art and sculptures to the art lovers and general public. This Mobile application removes the barrier of restrictions imposed on people and let them relish over the art presented by different artists or artistic places.

2. General Description

2.1 Product Perspective

The product must require the registration and login feature to authenticate the details of target users and should provide the authorization access to the respective target users. The product must offer the privacy and security to the users. it's getting difficult for the users to identify and explore the art galleries which are near-by. The product nearby display feature will lists the nearby art galleries and museums more accurately which will be greater advantage for the users to explore the art.

2.2 Product Functions

The product functions for registration feature are sign up and log in. The user details like full name, e-mail, mobile number, address, username, and password are required for signup function and the data of these user details stores in the database. The login function is used to authorize the users for the application based on the user details which are acquired in registration.

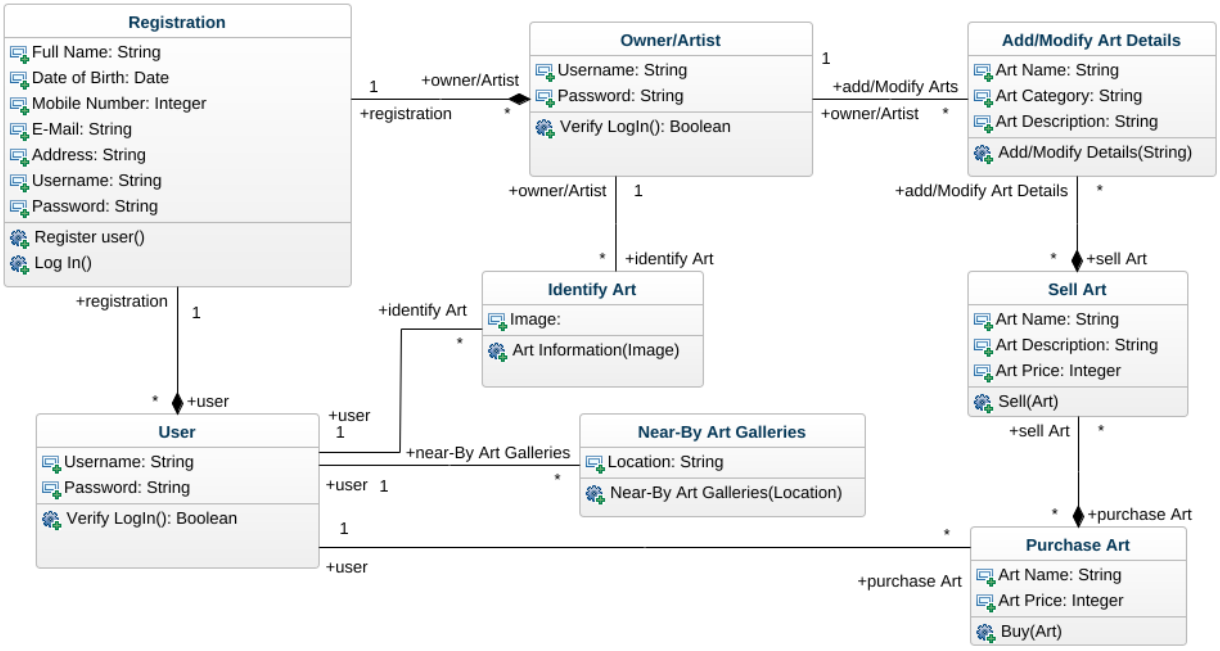
The search function will allow users to provide input location to compute the distance between the art galleries and input location to lists which are nearer. The nearby function will directly list the art galleries based on computation of distance between the user's location and art galleries.

2.3 User Classes and Characteristics

The user classes of the application are registration, user, owner/artists, database, and near-by art galleries. The registration and database have the attributes of full name, date of birth, mobile number,

e-mail, address, username, and password with their respective datatypes. The registration user class is required for both user and owner/artist class. The database class is required for all the other classes.

The registration has the operation as register user () which provides authentication to sign up and store the details in the database. The user and owner/artists use verify login () operation to provide the authorization for the respective target users based on the registration details in the database and the relation for user and owner/artists to registration is one to one. The nearby art galleries class have the attribute of location to perform the nearby art galleries () operations to display the listing of art galleries which are nearer to the input location.



UML Class Diagram

The user characteristics in the application are

End -User:

Users can be able to sign up through the application and the registration details are stored in the database. once the user details are recorded in the database, the users can be able to login with the credentials and view the list of art galleries and museums which are nearer.

Art Gallery Owners:

Art gallery owners must select the checkbox as owners while registration. The owners can be able to login to the application with the credentials once the registration data is recorded in the database.

Artists:

While registration, Artists must select the checkbox as artists and signup the registration. Once the records are added into database, the artists can be able to login to the application. the artist can find the listing of nearby art galleries and museums.

2.4 General Constraints

The software should also be feasible with the ARM processors for ARM V-7 and ARM V-8 versions. The owner and artist accounts must be flexible to manage the art details while presenting the art. the Preferable backend database or any other cloud databases mainly due to its security and scalability. The developers should also be responsible for maintenance of the application after deployment for a specific period.

2.5 Assumptions and Dependencies

If The developers use the premium services of the IDE like Android studio and IntelliJ Idea for the application due to this the cost estimates will be higher. If the business owners prefer the artyfy private servers or third-party private database servers to store on behalf of the cloud-based database servers, then the design must be changed.

3. Specific Requirements

3.1 Functional Requirement - Registration and logins

3.1.1 Introduction

Most of the applications require registration and logins to identify and classify the users. In our application the target users classify as users, gallery owners and artists to access their respective interfaces.

3.1.2 Inputs

The input details of Users, Gallery owners and Artists which are required for registration are full name, date of birth, mobile number, email-id, address, username, and password.

3.1.3 Processing

In the process of registration, the details of users, gallery owners and artists are going to be validated for contents in email-id which contains '@' character and ends with '.com', phone number which contains 10 numeric digits, date of birth format(dd-mm-yyyy) and password which contains numerical, capital letters, small letters, and special characters. After validation these details stores in the database. The user's information which exists in the database are used for authentication purposes during logins.

3.1.4 Output

The users, gallery owners and artists will display the message whether they successfully registered to the application or not. The successful login redirects the users, gallery owners and artists to their respective interfaces.

3.2 Functional Requirements - List of nearby museums and art galleries

3.2.1 Introduction

Now-a-days users are highly considering the near-by feature in comparison. This feature is highly recommended when people are travelling to new places or residing in unknow places. In this feature, the users can be able to find nearby art galleries and museums from their respective location.

3.2.2 Inputs

The user's location act as an input for displaying the listings of nearby art galleries and museums. The users can provide input location as a text and gets the display of listings from that user particular location.

3.2.3 Processing

Based on the input location, the application fetches the details of art galleries and museums from the database. Using searching techniques, it can be able to identify the art galleries and museums based on locations and display the listings of art galleries and museums using sorting techniques according to the distance between the art galleries and users respective location.

3.2.4 Output

The users get the output display of nearby art galleries and museums in the order based on their respective location or preferred location. The application also displays the details of that art gallery as art gallery name, art gallery picture and location.

3.3 Functional Requirements - Specify the registration of museum for Art Owner

3.3.1 Introduction

Whenever, a user chooses to be an art owner, they need to feed their museum details to the system to present their arts through the application.

3.3.2 Inputs

Owners need to choose a picture and should provide details like name, description, and location (residing city and coordinates).

3.3.3 Processing

Basing on inputs details which was given by art gallery owner or museum owner, the details are automatically saved into museums list.

3.3.4 Output

Owner can now be able to add art and those will be visible to end users.

3.4 Functional Requirements - Specify Add, Edit and Delete art by art owner

3.4.1 Introduction

We would like to implement all the mentioned transactions related to Artwork such that user can easily get UI to perform such operations. All the information related to artwork is fetched from the database through specific interfaces assigned for the tasks only, this helps user to find, store, remove and edit information about any new artwork or existing ones.

3.4.2 Inputs

Users add required information to add, delete, update and fetch information from database. For adding information related to art, user needs to add basic details such as name, phone number, email, etc. For fetching details, user adds art id and the application will fetch information further from database. For deleting details about art, user adds art id to delete the record from database. For update any detail, user adds information in form, and it gets updated in database.

3.4.3 Processing

After receiving information from user, the application start processing the contents. For add operation, application receive details and transfers it into relevant interface to save it further to database after applying appropriate logic. For update, delete and fetch operations, similarly application receives information and apply relevant logic to make the data sufficient to be saved in the database, which can be used in future.

3.4.4 Output

After adding data for add operation in form, user gets a unique art id which will be used for future reference. For other transactions, user will be notified for the successful or unsuccessful operations performed.

3.5 Functional Requirements - Art Finder feature

3.5.1 Introduction

One can use the art finder tool to capture any image that fascinates you using your phone's camera or upload an image from your directory to our app. In a comparable art feature, we send this image to our machine learning method. This function is incredibly useful for art enthusiasts because it allows them to take a picture at any moment and acquire more information about it in seconds. The similar art feature is the second feature. The ARTYFY app detects the similar art designs and patterns of the captured image and compares it with the images of the various arts available in the database. If similar images are found then, the details regarding similar images like artist name, art style, location of the image, cost of the Art, and other additional details.

3.5.2 Inputs

For Art finder feature we use the camera API. The Android framework supports capturing images. Here are the relevant classes which are android.hardware.camera package is the primary API for controlling device cameras. It can be used to take pictures or videos when you are building a camera application. An intent action type of mediastore_action_ image_ capture or mediastore.action_ video_capture can be used to capture images or videos without directly using the camera object.

3.5.3 Processing

The users upload the art image int the art finder application. based on the uploaded art image, the application uses image recognition feature for the identification of the art and the image classification feature to provide the art details to the user.

3.5.4 Output

The users and gallery owners will be able to view the details of the art using the art finding feature and compare the cost and other details of the paintings from the database. If the painting style is not available in the database, then new entries will be created for new paintings to be stored in the database.

3.6 Functional Requirements - View Arts and Paintings

3.6.1 Introduction

This function will allow the users to view the arts and paintings which the museum owners or the art gallery owners uploaded.

3.6.2 Inputs

The input for art details which are art image, art name, art description, art price was provided by the art gallery owners under their respective art galleries or museums.

3.6.3 Processing

The details of arts provided by the art gallery owners are stored in the database respective to their art galleries. In the listing of art galleries based on the nearby feature, the users can view all the arts and paintings in their respective art galleries.

3.6.4 Output

The user can view the details of art images, artist name, art description and art price in the respective art galleries which was maintained by art gallery owners.

3.7 Functional Requirements - Payment feature

3.7.1 Introduction

This feature is only available when the art owner wants to sell his art item then user will see a buy option for the that item respectively. Payment is done by third party payment services. This is a low-level feature which can be release at last sprint and needs a thorough testing as it need to be much reliable and accurate in real time application.

3.7.2 Inputs

Intended user who wants to buy an available art, clicks on buy button.

3.7.3 Processing

The control transfers to payment service and then returned to our current system.

3.7.4 Output

If the purchase is successful, then item will be sold and moved to purchase catalogue, if not successful item is still under selling.

3.8 Functional Requirements - List of arts based on artists

3.8.1 Introduction

This feature provides the list of arts to the users based on the artist’s name. The users can easily view their favorite artist arts based on the artist’s name.

3.8.2 Inputs

The art gallery owner provides the artist’s name input details when creating the arts respective to their art galleries. This feature takes artist name as input to list the arts based on the name of the artist.

3.8.3 Processing

While creating the art in their respective museums, the art gallery owners provide the artist's name to the respective arts. Based on the artist's name which was provided by art gallery owners the list of arts based on artist feature list the arts based on artist name to the users.

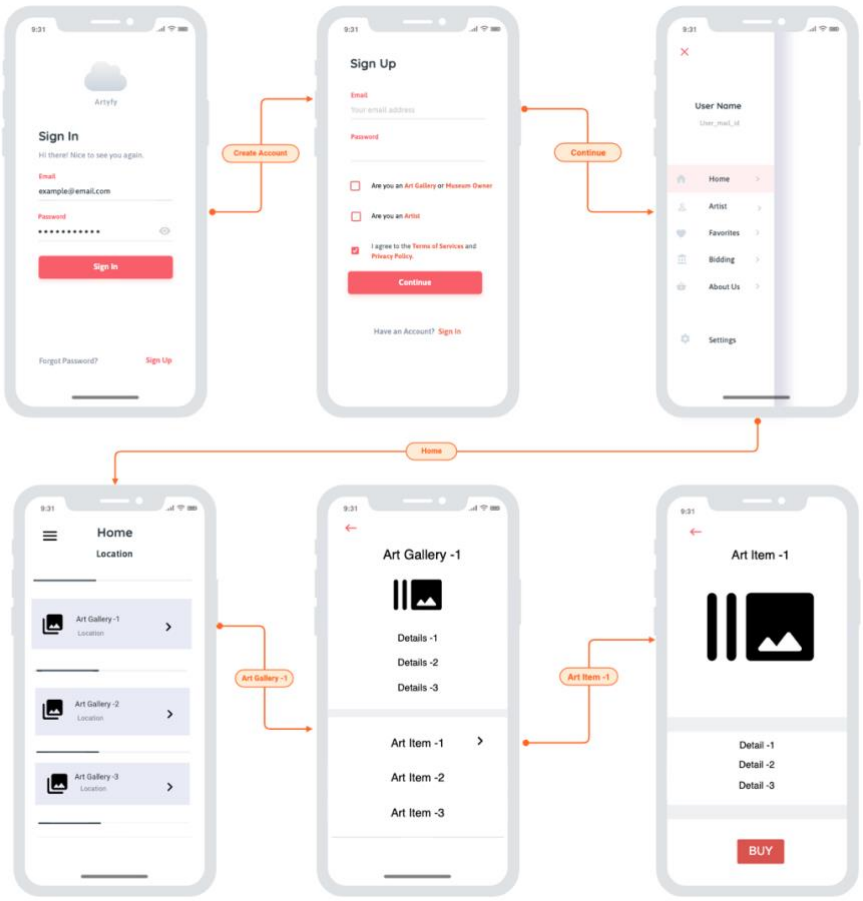
3.8.4 Output

The users get the list of arts based on the artist names which was provided by art gallery owners when they are adding arts for their respective art galleries.

4. External Interface Requirements

4.1 User Interfaces

The user interface for List of nearby museums and art galleries features are
Displays the listings of art galleries and museums based on user's location.
Based on user search input, the application displays the art galleries.



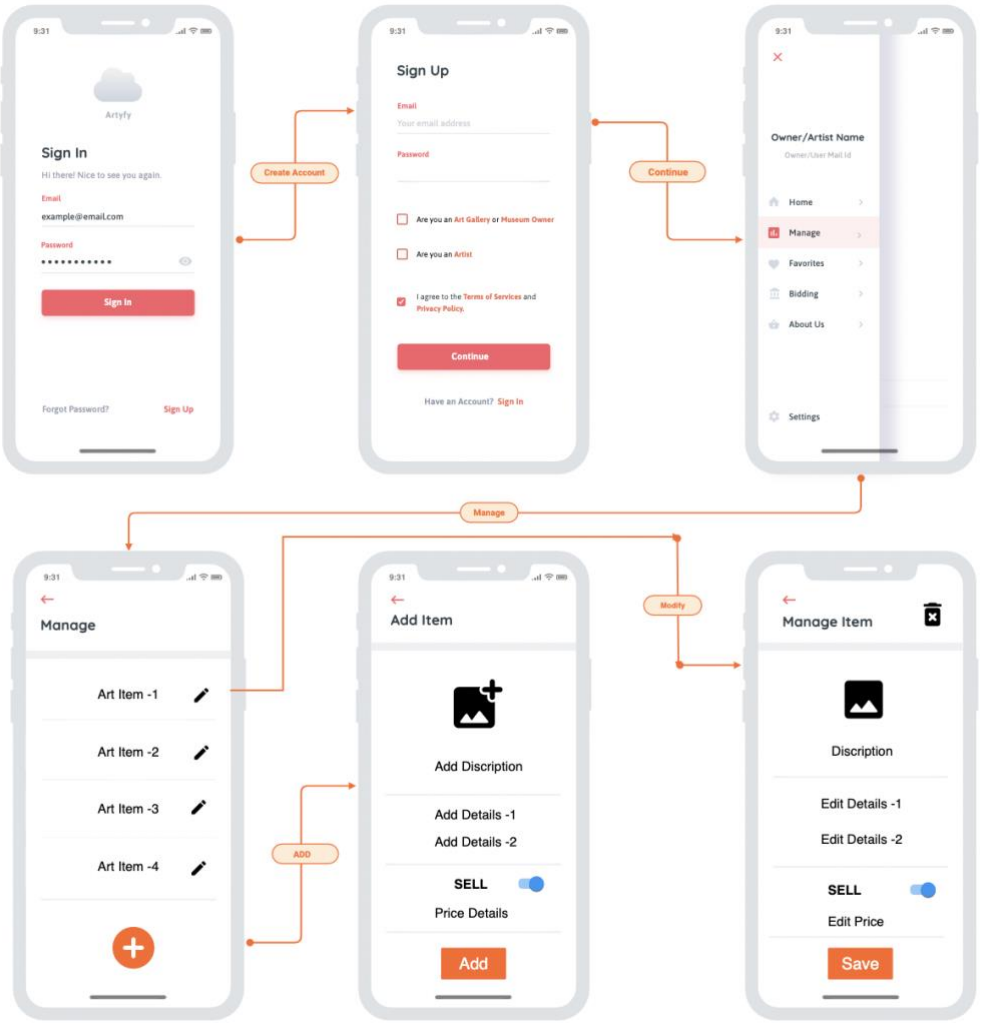
The user interface for Registration and logins feature are

Form based registration for new users. In the form-based registration the attributes are

- Full name
- Mobile number
- E-mail
- Location
- Username
- password

Form based log-in for registered users.

The art gallery owners have the feature to manage the art respective to their museums and the users have access to purchase the art when the art gallery owner set an option to sell the art.



4.2 Hardware Interfaces

The hardware configuration is ARM, and the application is also suitable for ARM-V7 and ARM-V8.
Accelerometer, Magnetometer and Gyroscope sensors are used to identify the user's location in the application.

4.3 Software Interfaces

The software interfaces for the application are

- Android studio
- XML
- JAVA
- Firebase

The application is suitable for Android 8.0 and above versions.

4.4 Communications Interfaces

The user or product owners or artists received a message notification after their successful registration through email.

5. Other Non-functional Requirements

5.1 Performance Requirements

The application performs exemplary user experience and standards in order to achieve successful registration and maintain scalability throughout the registration process. The execution time, response for the user data that is stored in the database is responsive and quick. As the application is used by the end users, art gallery owners and artists, the application should provide the better performance throughout the registration and login. The application responsive time should be faster for listing the nearby art galleries while fetching the information from the database.

5.2 Safety Requirements

To maintain quality assurance and quality control of the application we must perform monitoring and reviewing of the code and the application programming interfaces and test the database frequently in order to avoid connection time out. The application ensures to maintain the connection availability between the application and the database. The data cannot be accessed or modified by users other than owners and artists. By frequently monitoring the code reviews will helps to maintain the quality and efficiency of the application.

5.3 Security Requirements

The user data in the application is authenticated and secured through premium services and also maintain confidentiality for the user data as it is maintained by museums owners and the application will also specify the security requirements of the user in terms and conditions before submitting the user registration form. The application provides authentication and authorization in the registration and login tasks. The authentication will identify the users and the authorization will provide access based on users, art gallery owners and artists. The art information is maintained by art gallery owners or artists and only the authorized users can fetch the art information.

5.4 Software Quality Attributes

The users experience good performance and availability while logging in to the application and access the exact locations that are prefered by the users. The software quality in handling the complexity and performance of the application is maintained by examining the amount of time the users are using the app. The end users experience the high availability and better performance while registering and login tasks and the application can be reliable and compatible, it will minimize the network latency and maximize the throughput across the application.

6. Other Requirements

6.1 Database Requirements

The user data is stored in the database with regular backups. The connections and storage to the database are monitored frequently through an alarm in order to avoid timeouts. The application must perform database backups to safeguards the target user data. The performance tuning activities will help to improve the database performance and decreases the operation time while retrieving the data from the database.