GitHub Username: irahavoi

QRiosCat

Description

QRiosCat is a QR code scanner for museums and art galleries. It allows to quickly retrieve interesting facts about your favorite artworks by scanning qr-codes located near exhibits, add personal notes and build your personal collection of all artworks that you like. Earn bonus points every time you add a new artwork to your collection and use points to get free museum admissions and discounts!

Application is built using a client-server architecture where the server side is responsible for storing and providing content describing each artwork and the client application allows museum visitors to retrieve the content by scanning QR codes, store the history of favorite exhibits, make notes and get bonus points with each scan.

Intended User

Museum and art gallery lovers.

Features

- Scans QR Codes assigned to museum exhibits
- Fetches exhibit description from a web service
- Stores user personal notes about saved artworks
- Allows users to earn bonus points with each scan and trade these points for free museum admissions and discounts.

User Interface Mocks.

Screen 1

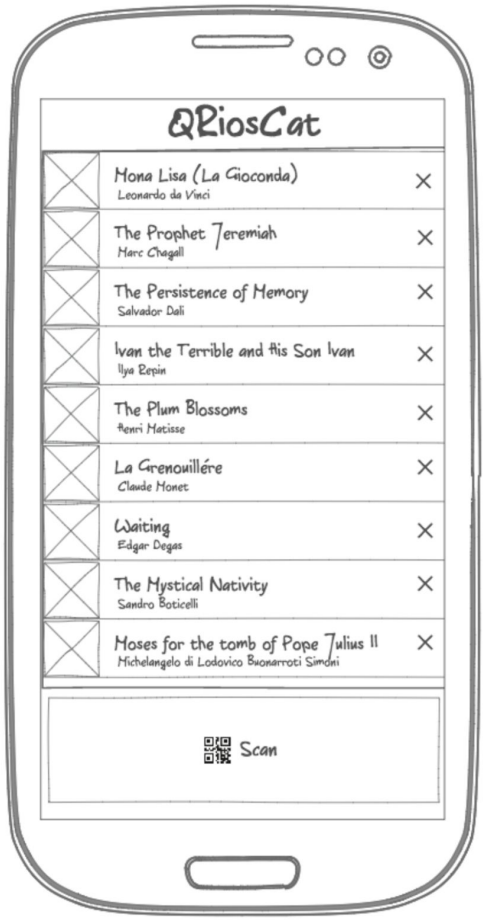
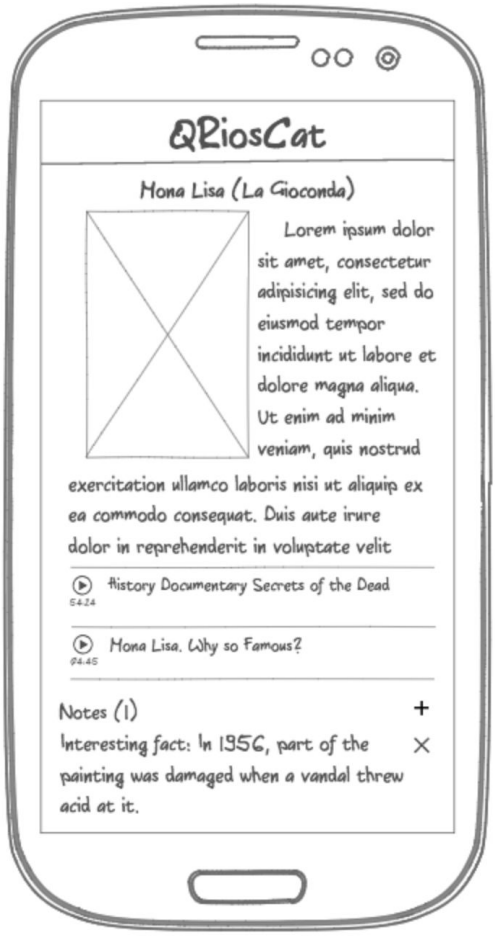


Exhibit List view. Contains a list of already scanned items and a button for scanning new exhibits. User can delete existing items, scan new items as well as select an item to navigate to the details view.

Screen 2



Details view. Contains an image of the exhibit, its' description and optional links to youtube videos related to the exhibit. User is able to add/edit/remove personal notes on the selected exhibit.

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

Exhibit data will be stored in the local SQLite database and fetched using a Content Provider.

Describe any corner cases in the UX.

There are three main views: exhibit list view, exhibit detail view, and QR-code scan view.

List view allows user to scan new artworks, remove and navigate to existing artworks. Exhibit detail view provides text description of the artwork, optional youtube links to videos related to the artwork and user's personal notes. QR-code scan view allows user to capture the qr code using the camera of the user's device.

Describe any libraries you'll be using and share your reasoning for including them.

Picasso - for image loading and caching
Google Play Services - for QR code scanning and analytics.
Retrofit - for RESTful communication with the web service providing exhibits data.

Next Steps: Required Tasks

Task 1: Project Setup

- Configure Manifest file
- Configure build file - add dependencies

Task 2: Implement UI for Each Activity and Fragment

- Build UI for exhibit list activity
- Build UI for exhibit details activity
- Build UI for QR code scan activity

Task 3: Exhibits web service implementation

- Implement a web service providing data for exhibits
- Implement a mock "museum" page containing several generated QR-codes for various exhibits

Task 4: Scanning an Exhibit

- Implement QR-code scanning functionality

Task 5: Communication with the Web Service

- Implement fetching exhibit data from the web service using the id received as a result of qr-code scanning

Task 6: Data persistence

- Implement storing scanned exhibit data on the client device using content providers and sqlite