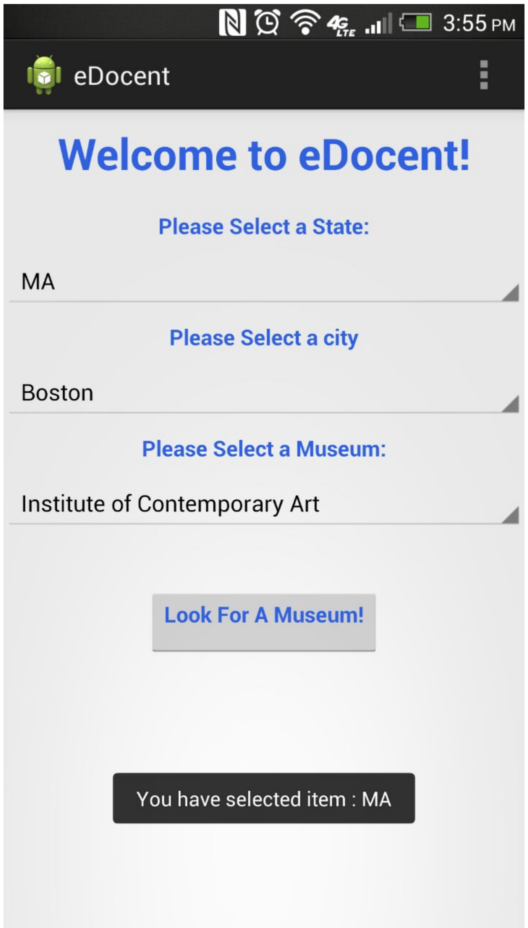
eDocent Android Application Design Documentation

User Interface

Splash Page

In the splash page, the user is shown three spinner objects, drop down menus. The first drop down menu will prompt them to select a state, the second a city/town, and finally the third will prompt them to select a museum that will be in the state and city/town they selected. All these menus will be dynamically generated by API calls. Once the user selects a museum, they will be directed to that museum's page in the Android application.



Welcome to eDocent!

Please Select a State:

MA

Please Select a city

Boston

Please Select a Museum:

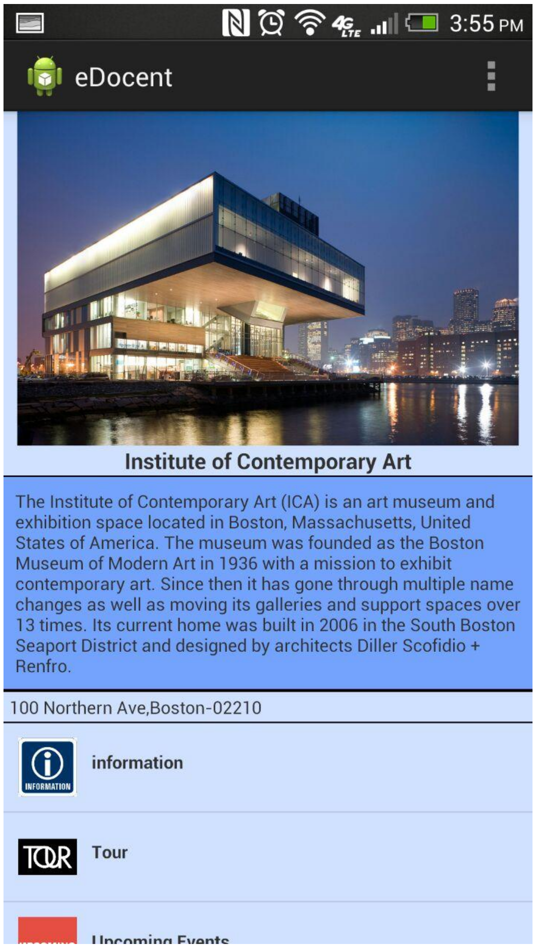
Institute of Contemporary Art

You have selected item : MA

Fig1. Splash Page

Museum Home Page

The museum home page will display a picture of the museum, a small blurb about the museum, information such as museum hours, address, tour, upcoming events, spotlight of the week, collections, maps, QR scanner etc. If they click on collections, they will be able to view the art that the museum currently has. Users can learn more about specific art pieces at the museum by clicking on them under the collections list. This information will be given by the museum.



Institute of Contemporary Art

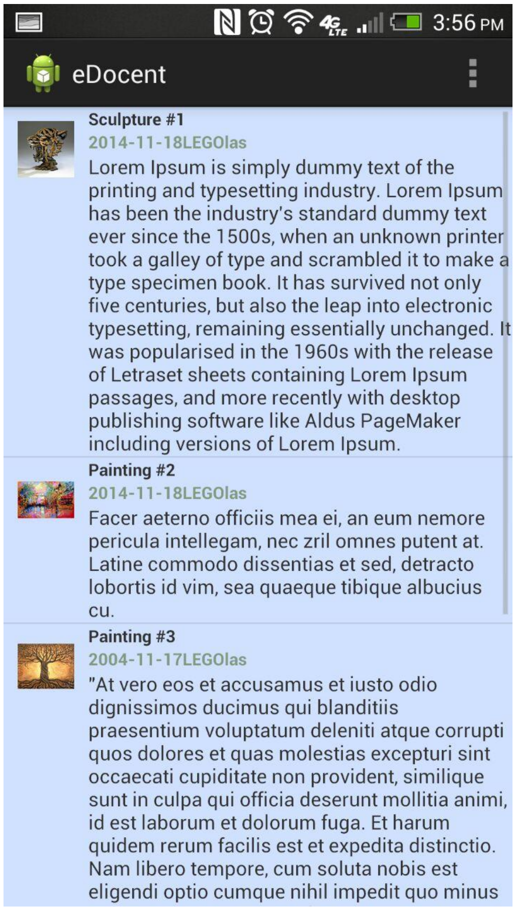
The Institute of Contemporary Art (ICA) is an art museum and exhibition space located in Boston, Massachusetts, United States of America. The museum was founded as the Boston Museum of Modern Art in 1936 with a mission to exhibit contemporary art. Since then it has gone through multiple name changes as well as moving its galleries and support spaces over 13 times. Its current home was built in 2006 in the South Boston Seaport District and designed by architects Diller Scofidio + Renfro.

100 Northern Ave,Boston-02210

Fig2. Museum Home Page

Collections Page

The collections page displays the list of artworks that belong to the Museum we selected. We get a small image of the art, date of artwork, artist and the art information.



Sculpture #1
2014-11-18LEGOlas
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Painting #2
2014-11-18LEGOlas
Facer aeterno officiis mea ei, an eum nemore pericula intellegam, nec zril omnes putent at. Latine commodo dissentias et sed, detracto lobortis id vim, sea quaeque tibique albucius cu.

Painting #3
2004-11-17LEGOlas
"At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus

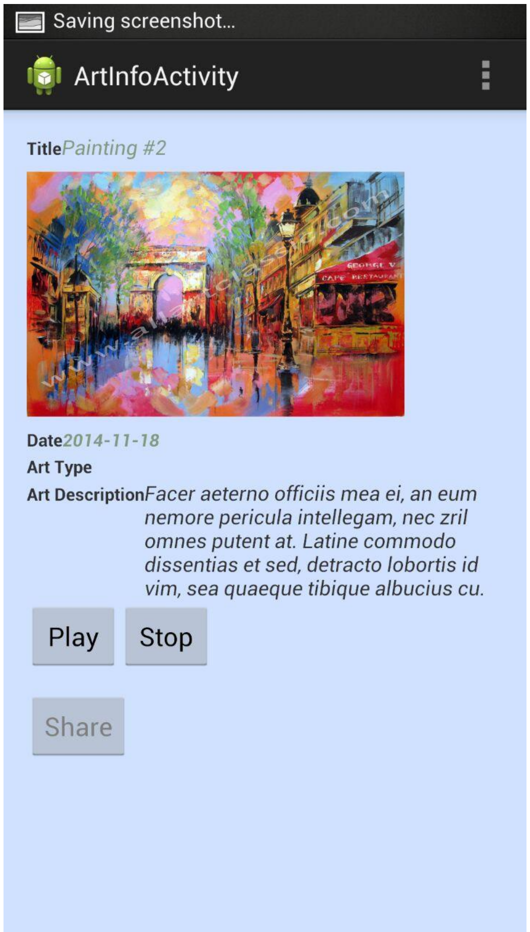
Fig3. Collections Page

Art Page

Art pages are what display information about a specific piece of art in the museum. Users will be able to read more about the piece of art than what is displayed at the museum. Each art piece will also include the art type, period in which it was made, art history. This information will be given by the museum. The art page also has a Play and Stop button which facilitates the audio of the history which the curator would need to upload. A share button helps share the information on social media, however this feature has not been extended and does not work completely, at this point we can login to the social media page a browse.

In the Android application, the user will only be giving input in the homepage. Otherwise, they are only able to see whatever the museum uploads and shows them. They cannot edit any museum or art information within the mobile application.

To get the share option we used *SocialAuth* library - Android SDK allows integrating with several social networks. This SDK contains the java libraries that do the OAuth as well as the REST calls for each social provider.



Saving screenshot...

ArtInfoActivity

Title*Painting #2*

Date*2014-11-18*

Art Type

Art Description*Facer aeterno officiis mea ei, an eum nemore pericula intellegam, nec zril omnes putent at. Latine commodo dissentias et sed, detracto lobortis id vim, sea quaeque tibique albucius cu.*

Fig4. Art Information Page

Maps

This feature helps us connect to google maps and get the location of the museum from our current location. However, make a call feature does not work.

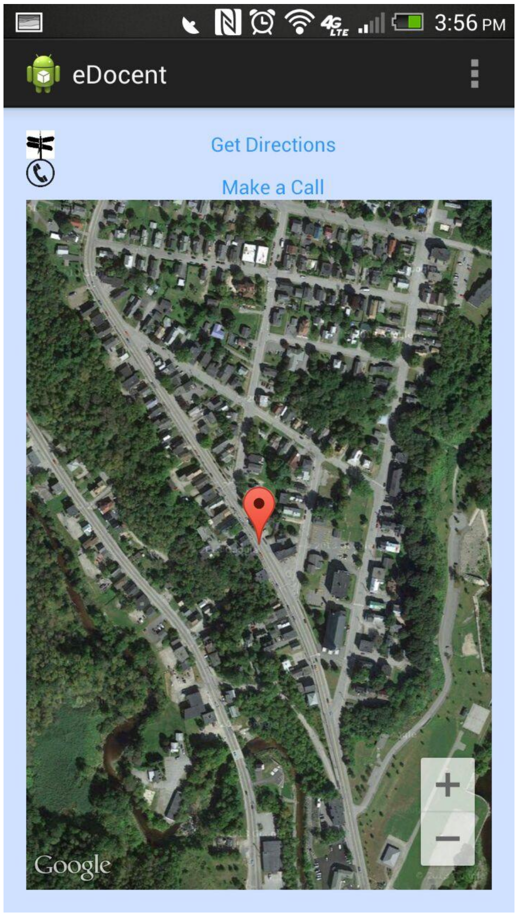
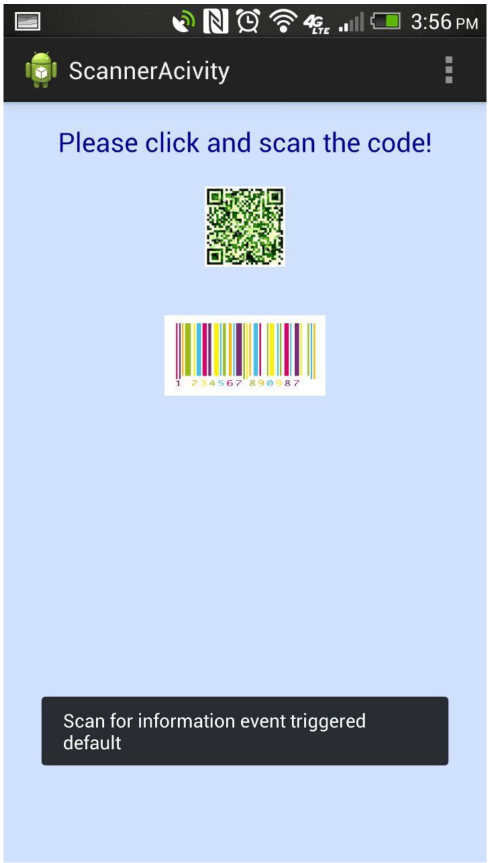


Fig5. Maps Page

QR Code Scanner

This feature facilitates scanning of code, when the scanner identifies the code, it displays the relevant art information details on the application. To support this feature we used *zxing* which is a library which supports decoding and generating of barcodes (like QR Code, PDF 417, EAN, UPC, Aztec, Data Matrix, Codabar) .



Please click and scan the code!

Scan for information event triggered default

Fig6. QR Code Scanner Page

Other features

We wanted to include other features from the Museum Home Page such as Information, Tours, Upcoming Events etc, however lack of time we could not work more on these, However we do hab provision in the backend database to enter these details and also some basic code from android, however it is not complete. We focused more on the Museum and Art information along with Audio, Maps, QR Code and Sharing features.

Behind the Scenes

All the information the mobile application needs is stored in the database. The mobile application calls an API, which queries the database and gives the requested data to the mobile application. The data is in JSON, therefore, the mobile application parses the data and then displays the information to the user.

When an application is launched, we call a GUI thread for the user view. We do not call the server to retrieve data using the same thread, if we would use the same thread, the application will hang. Hence we create a new background thread to fetch data from the server.

Architecture

We use the Model-View-Controller architecture for implementing the user interface. In the below UML diagram we give a brief description o

f the controller activities. The GenericSeeker class is the base class which contains the HTTPclient and the JsonParser. Each page would have its own controller. For example the Museum Splash Page Client class would fetch data from the database. The Splash Page Controller has a pointer to the data model. The data model is populated and we can view the information in the user interface.

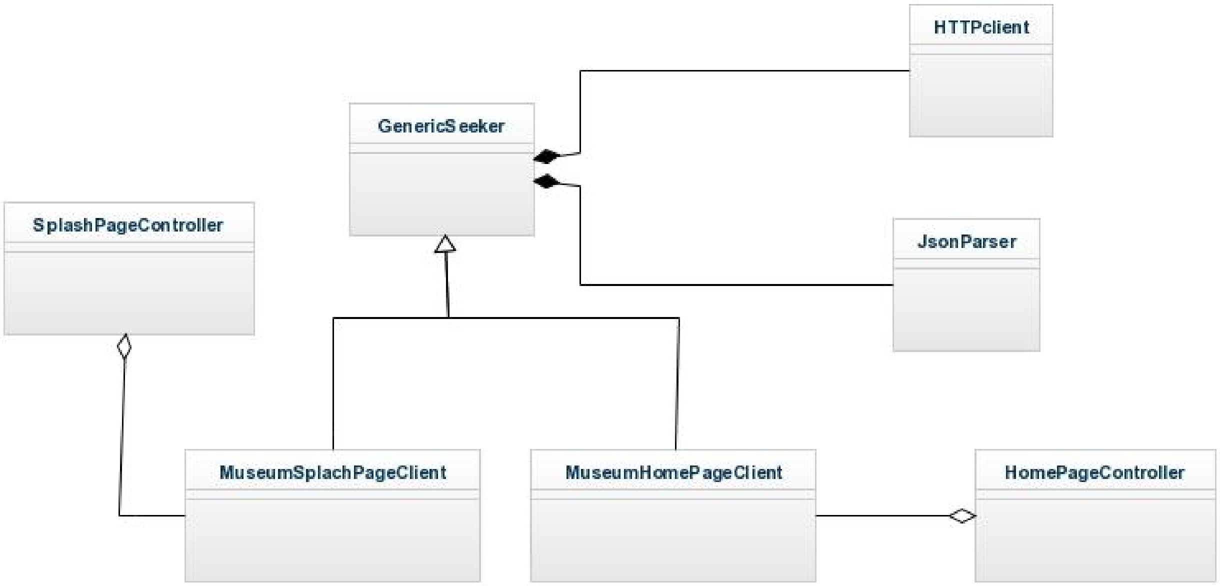


Fig3. Controller UML Diagram