

eDocent

Group 5 - Iteration 2

Overview

- Iteration 2 Goals
- Web App Testing and Demo
- Android Application Demo
- Iteration 3 Goals

Iteration 2 Goals

- Museum user can create instance of art “object” and Android user can access collection
- QR Reader on Android Phone
- Registration
- Stretch goals
 - Social Media Sharing
 - Map directions

Django administration

MuseumCurator / Museums / Add museum

Add museum

Name:

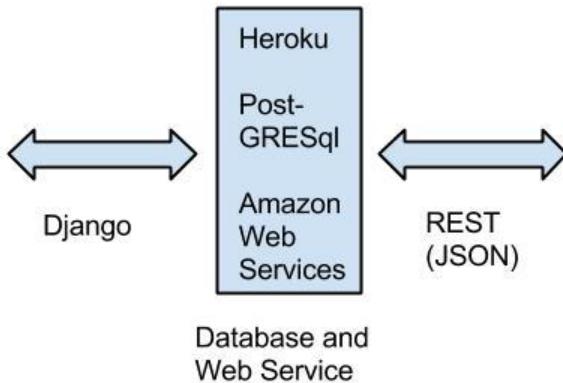
Location:

Hours:

Website:

Tour times:

Collections:



Web Application

Android Application

Backend - Registration

- Signing up process
 - user visit/account/register
 - Filling the form
 - Clicking on submit button
 - Activating the account
 - Museum user: via email
 - Admin: manually after validation

127.0.0.1:8000/accounts/register/

Create an new Museum account

Username:

E-mail:

Password:

Password (again):

Museum name:

Street Address:

City:

State:

ZipCode:

Opening Hours Monday:

Opening Hours Tuesday:

Opening Hours Wednesday:

Opening Hours Thursday:

Opening Hours Friday:

Opening Hours Saturday:

Opening Hours Sunday:

events:

Description:

Backend - Registration

membership:

ticket_prices:

parking:

Website:

visitor_info:

image: No file selected.

- Registration Form (accounts)

Testing with Selenium and Demo

```
def test_can_create_new_museum_via_admin_site(self):
    # Gertrude opens her web browser, and goes to the admin page
    self.browser.get(self.live_server_url + '/admin/')

    # She sees the familiar 'Django administration' heading
    body = self.browser.find_element_by_tag_name('body')
    self.assertIn('Django administration', body.text)

    username_field = self.browser.find_element_by_name('username')
    username_field.send_keys('admin')

    password_field = self.browser.find_element_by_name('password')
    password_field.send_keys('admin')
    password_field.send_keys(Keys.RETURN)

    body = self.browser.find_element_by_tag_name('body')
    self.assertIn('Site administration', body.text)
```

```
museum_links = self.browser.find_elements_by_link_text('Museums')

self.assertEqual(len(museum_links), 1)

museum_links[0].click()

body = self.browser.find_element_by_tag_name('body')
self.assertIn('0 museums', body.text)

# She sees a link to 'add' a new poll, so she clicks it
new_museum_link = self.browser.find_element_by_link_text('Add museum')
new_museum_link.click()

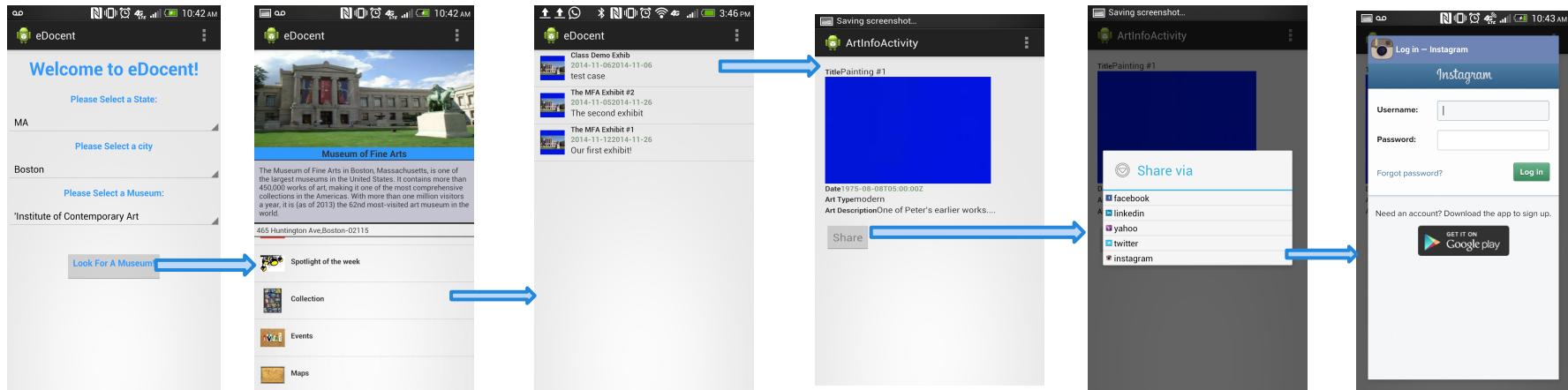
body = self.browser.find_element_by_tag_name('body')
self.assertIn('Name:', body.text)
self.assertIn('City', body.text)
self.assertIn('State', body.text)
```

Creating an Art Object Demo

- Creating and uploading associated image for art object
- Creating a collection object for Android user to view

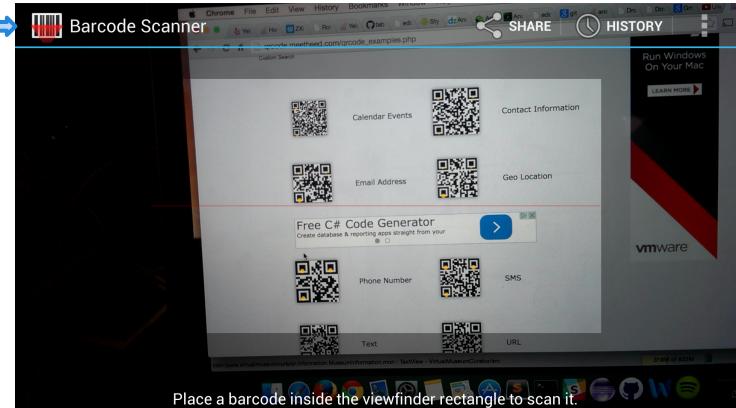
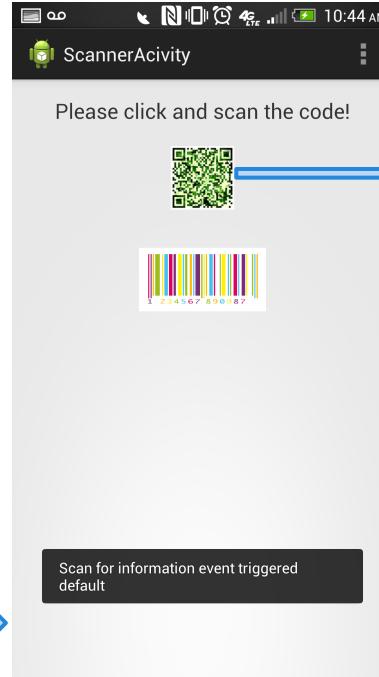
Android

Displaying collections of art objects



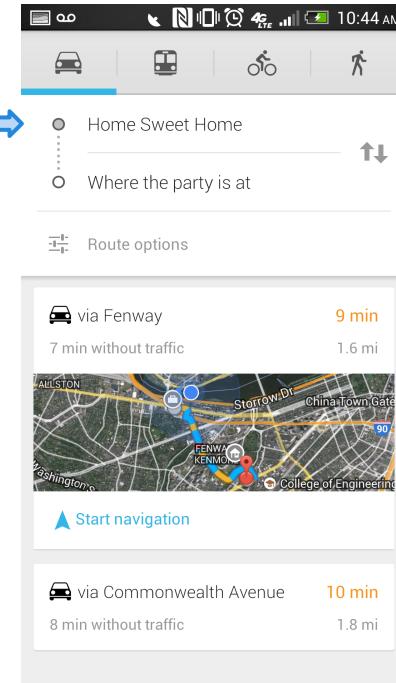
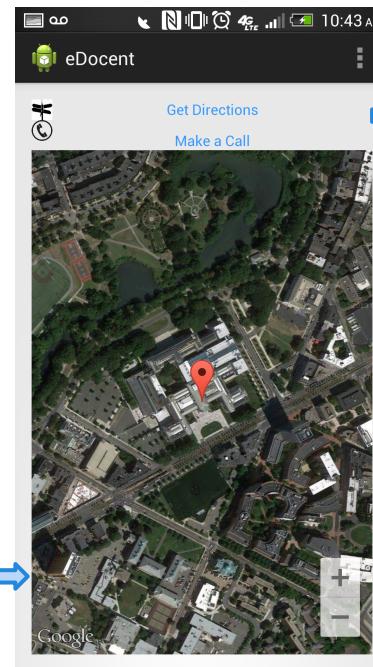
Android

The QR scanner



Android

Map and directions to the Museum



Android Demo

- <https://www.youtube.com/watch?v=jMxbKi5ARc0>
- **zxing**- A library which supports decoding and generating of barcodes (like QR Code, PDF 417, EAN, UPC, Aztec, Data Matrix, Codabar)
- **SocialAuth** -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

Iteration 3 Goals

- Get relevant images for museum home page and art work page
- QR codes should directly lookup pieces of art
- Complete testing

THANK YOU !