

Project Herodot

*Digitizing Heritage Before It Becomes Ancient*

Cultural Heritage Content Management System with Built-In Annotations Feature

AyranIsTheNewRaki aka Group 1| SWE 574 | 29 May 2017

Table of Contents

[1. Introduction 3](#_Toc484606152)

[2. Abbreviations 3](#_Toc484606153)

[3. Abstract 4](#_Toc484606154)

[3.1. On Cultural Heritage Content Management Systems & Digitization of CHO 4](#_Toc484606155)

[3.2. Europeana Data Model and Structure: 6](#_Toc484606156)

[3.3. Research on Annotations 12](#_Toc484606157)

[Hypothes.is utilized as annotation benchmark 21](#_Toc484606158)

[4. Project Plan 25](#_Toc484606159)

[5. Milestone Reports 26](#_Toc484606160)

[5.1. Milestone Report #1: Project Management Decisions (Closed on 20th of Feb) 26](#_Toc484606161)

[5.2. Milestone Report #2: Contributing Guidelines Documentation (Closed on 4th of March) 28](#_Toc484606162)

[5.3. Milestone Report #3: Analysis Document (Closed on 8th of March) 28](#_Toc484606163)

[5.4. Milestone Report #4: Project Planning And High Level Architecture Formation (Closed on 3rd of Apr) 30](#_Toc484606164)

[5.5. Milestone Report #5: Sanity Check with Registration Prototype (Closed on 10th of April) 32](#_Toc484606165)

[5.6. Milestone Report #6: Create System V1 (Closed on 17th of April) 33](#_Toc484606166)

[5.7. Milestone Report #7: View System V2 (Closed on 20th of April) 34](#_Toc484606167)

[5.8. Milestone Report #8: Edit System V3 (Closed on 27th of April) 36](#_Toc484606168)

[5.9. Milestone Report #9: Iteration V4 (Closed on 20th of May) 37](#_Toc484606169)

[5.10. Milestone Report #10: Iteration V5 (Closed on 27th of May) 37](#_Toc484606170)

[5.11. Milestone Report #11: Iteration V6 (Closed On 29th Of May) 38](#_Toc484606171)

[6. Requirements Elicitation 39](#_Toc484606172)

[7. Software Design Document 42](#_Toc484606173)

[7.1. High Level Entity Relationship Diagram 42](#_Toc484606174)

[7.2. System Component Diagram for Web 43](#_Toc484606175)

[7.3. System Component Diagram for Mobile 43](#_Toc484606176)

[7.4. System Sequence Diagram for Web 44](#_Toc484606177)

[7.5. Mobile Sequence Diagram 46](#_Toc484606178)

[7.6. System Deployment Diagram 46](#_Toc484606179)

[7.7. Mock-up Diagrams 48](#_Toc484606180)

[7.7.1. Web Side Mock-Ups 48](#_Toc484606181)

[7.7.2. Mobile Side Mock-Ups 50](#_Toc484606182)

[8. System Specifications Document 56](#_Toc484606183)

[8.1. System Specifications Summary 56](#_Toc484606184)

[Mobile (Android): 56](#_Toc484606185)

[Front-End: 56](#_Toc484606186)

[Back-end: 58](#_Toc484606187)

[8.2. System Specifications Details 59](#_Toc484606188)

[Deployment Configurations of Front-end Application to Nginx on Debian 61](#_Toc484606189)

[9. Test Documents 62](#_Toc484606190)

[10. Tools and Technologies 65](#_Toc484606191)

[11. References 66](#_Toc484606192)

# Introduction

The project is developed as a part of SWE 574 - Software Development as a Team lecture of Boğaziçi University Software Engineering department. The project basically aims to create web and mobile applications where the users can create and browse cultural heritage objects and create annotations for the content in the applications.

The project life cycle consists of research, plan, design, development, test and documentation phases. The project members are Emre Bolat, Eren Erdogan, İdil Gün, Nur Ertem Unden, Onur Uygur, Taygun Gökdemir and İlker Yılmaz (was mostly absent due to his tight business schedule).

# Abbreviations

Below the reader could find abbreviations used throughout the paper.

* CHO (Cultural Heritage Item)
* UNESCO (United Nations Educational, Scientific and Cultural Organization)
* W3C (World Wide Web Consortium)
* EDM (Europeana Data Model)
* DCMI (Dublin Core Metadata Initiative)
* AWS (Amazon Web Services)
* E-R (Entity Relationship Diagram)

# Abstract

The project “Herodot” aims to provide users to maintain an annotative environment for specific heritage items. Below the reader could find the research work regarding the project content.

## On Cultural Heritage Content Management Systems & Digitization of CHO

UNESCO is a specialized agency of the United Nations primary resource and a standard setting authority for Cultural Heritage, similar to the main international standards organization that what is W3C is for the internet.

According to UNESCO cultural heritage contains several main categories of heritage as they stated below.

* Tangible Cultural Heritage,
* Movable Cultural Heritage (paintings, sculptures, coins, manuscripts),
* Immovable Cultural Heritage (monuments, archaeological sites, and so on),
* Underwater Cultural Heritage (shipwrecks, underwater ruins and cities),
* Intangible cultural heritage (Oral traditions, performing arts, rituals)
* Natural heritage (Natural sites with cultural aspects such as cultural landscapes, physical, biological or geological formations, heritage in the event of armed conflict. )

After a careful analysis of UNESCO web sites and resources provided for categorization of the Cultural Heritage items a draft category of cultural heritage items were determined. (UNESCO.ORG, n.d.)

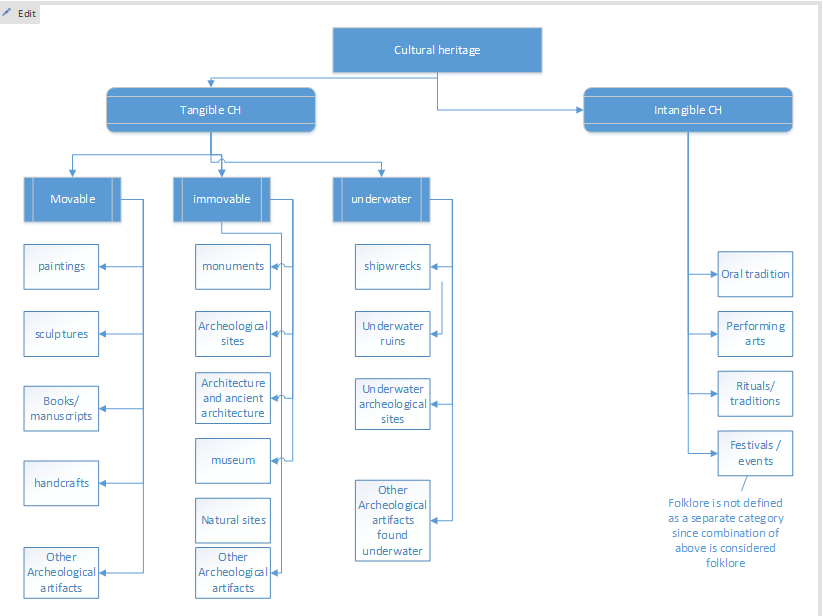


Figure 1 CHO Categories from UNESCO

Categorization of Cultural Heritages was the starting point for digitization of CHO. However, there were many more standards, data structures, schemas to be researched about. Having researched various Cultural Heritage Digitization attempts, the one we focused on at the end; which is also the only one listed under cultural heritage category from the w3.org’web site as a use case is: Europeana. (Europeana as a annotation use case, n.d.) The use of linked open data, JSON-LD, annotations, vast and detailed documentation on the project provided as a benchmark for our project.

After reading the Europeana Data Model, (Europeana Data Model Guidelines, n.d.) and linked data (Europeana Linked Open Data, n.d.) and its work area, the logic and technology behind semantic web and annotations and their data models and standards, it was evident that a schema was already in place and as software engineers we needed to understand and use these standards in order to deliver the requirements asked by the client.

The client would be satisfied to know that the end product/ web site with its data and metadata will be interoperable in web and may also have a value to be on top of other resources by the use of these linked in data benefits, being a pioneer in the new linked in data network. More inbound links means; better site score, enriched and visible data. Trustworthy linked open data being made available now, gives a big advantage in the www in terms of users, traffic, etc. by being a canonical reference point.

In short, after reading all the standards in place on the linked data structure, it was pointless to come up with a new database schema. As software engineers, we will be using already established standards in our domain according to our needs. We will be utilizing the below mentioned structure provided by Europeana, as part of our back-end solution.

In addition, we highly recommend the short video provided by Europeana Labs on linked open data: <http://labs.europeana.eu/api/linked-open-data-introduction>

## Europeana Data Model and Structure:

The EDM Mapping Guidelines –give guidance for providers wanting to map their data to EDM. They show which property relates to which class and contains definitions of the properties, the data types that can be used as values and the obligation level of each property. It also has an example of original data, the same data converted to EDM and diagrams showing the distribution of the properties amongst the classes. The full set of EDM classes and properties are being implemented incrementally and the Mapping Guidelines is the reference document showing which are currently available. (Europeana EDM Mapping Guidelines)

 The EDM object templates – a working document is a simple wiki listing that shows which properties apply to which class and states the data types and obligation of the values. These templates should be regarded as a work in progress however and may be out of step with the Guidelines. Please refer to the Mapping guidelines for the current set of classes and properties in use. (Europeana Data Model Object Templates)

 The core classes are:

• edm:ProvidedCHO - the provided cultural heritage object

• edm:WebResource - the web resource that is the digital representation

• ore:Aggregation - the aggregation that groups the classes together

 Here as a team, we will be focusing on the core class provided cultural heritage item as a real object.

Properties for EDM:ProvidedCHO; meaning the attributes of the Monalisa painting, but not the attributes of the digitized image of the Monalisa painting.

Before we proceed on more terminology, dc and dcterms needs to be clarified.

DCMI is an up-to-date specification of all metadata terms maintained by the Dublin Core Metadata Initiative, including properties, vocabulary encoding schemes, syntax encoding schemes, and classes. Among these metadata, below are the ones to be used in Heredot data base schema.  (DCMI Metadata Terms by Dublin Core Metadata Inititative, n.d.)

Mandatory CHO attributes decided to be included in Heredot Database are listed:

* dc:title or dc:description
* Dc:subject
* Edm:currentLocation
* Dc:date
* Dc:creator

Table 1 EDM main types and definitions selected for Heredot project

|  |  |
| --- | --- |
| property | definition |
| dc:title or dc:description | A description of the CHO. Either dc:description or dc:title must be provided. |
| dc:format | Use for the terms generally applied to indicate the format of the cultural heritage object <dc:format>paper</dc:format> |
| dc:identifier | An identifier of the original CHO.  <dc:identifier>RP-T-1952-380</dc:identifier> |
| dc:language | The language of CHOs. Best practice is to use ISO  639 two- or three-letter primary language tags5  . Repeat for multiple languages.  <dc:language>it</dc:language> |
| dc:source | A related resource from which the described resource is derived in whole or in part i.e. the source of the  original CHO. (Not the name of the content holder: for this see edm:dataProvider.)  <dc:source>Security Magazine pp 3-12</dc:source> |
| dc:subject | The subject of the CHO. One of dc:subject or dc:coverage or dc:type or dcterms:spatial or dcterms:temporal  must be provided  <dc:subject>trombone</dc:subject>  or create a reference to an instance of the Concept class  <dc :subject rdf:resource=“http://id.loc.gov/authorities/subjects/sh85137992"/> |
| dcterms:created | The date of creation of the CHO.  <dcterms:created>Mid 16th century</dcterms:created> or <dcterms:created>1584</dcterms:created>  or create a reference to an instance of the TimeSpan class  <dcterms:created rdf:resource=“http://semium.org/time/15xx\_3\_third”/> |
| edm:currentLocation | The geographic location whose boundaries presently include the CHO. If the name of a repository, building,  site, or other entity is used then it should include an indication of its geographic location.  <edm:currentLocation rdf:resource=“http://sws.geonames.org/2950159”/> (Identifier for Berlin) |
| edm:isRelatedTo | The identifier or name of a concept or other resource to which the described CHO is related. E.g. Moby Dick  is related to XIX Century literature. Cf dc:relation.  <edm:isRelatedTo>Literature</edm:isRelatedTo >  Or link to resource  <edm:isRelatedTo rdf:resource=“http://www.eionet.europa.eu/gemet/concept?cp=4850”/> |
| edm:type | The value must be one of the types accepted by Europeana as it will support portal functionality :  TEXT, VIDEO, SOUND, IMAGE, 3D. (For 3D, when applicable, use the value “3D-PDF” in dc:format )  <edm:type>IMAGE</edm:type> (upper-case)  <edm:type>3D</edm:type> (upper-case) |

Below is a tangible example is provided (Europeana Solid Example , n.d.) and can be viewed in detail:

6.2.1 Provided CHO 
In the original LIDO record, these properties can be found in the set related to the descriptive 
metadata allowing the identification of the object. These properties only describe the Cultural 
Heritage Object provided to Europeana, in this case a musical instrument. 
description: Brass; ligature fitting on bell 
section at joint; on main Slides _ Eel 1 kith one coil, to face 
Repair History: Main slide gossibly not original of slide section of joint is 
tapered, bell section joint for cylindrical tenon) _ 
identifier>9UEDIN fier> 
_geormes.org/30173S2/" / > 
trombone _ pitch: 
type mime—db _ 
type rdf 
type> 
</edm p rcwidedCHO> 
"Buccin trombone. 
"Circa 1840* 
Nominal pitch: 
dc:date 
dc:title 
edm : Pro vided CHO 
"UEDlN:21a 
dc:type 
s kos:Conc ept 
http;//www.mimo-db.eu/ 
InstrumentsKeywOrds/4378 
-Technical description: Brass; ligature 
fitting on bell section at joint; stockings on 
main slides. gell with one coil, angled to 
face forwards. Repair History: Main Slide 
possibly not original (tenon of slide section 
of joint is tapered, bell section joint for 
cylindrical 
dc:description 
edm :type 
•IMAGE" 
Figure. 3 Provided CHO (Not all properties are shown for readability) 

Figure 2 provided CHO example

europeana 
collections 
Return to Home / Item 
Collections 
Explore 
Exhibitions 
Blog 
O Media Metadata 
Title 
Description 
Classifications 
Time 
Provenance 
References And 
Relations 
Location 
Similar Items 
Entities 
OUR MISSION 
Buccin trombone. Nominal pitch: B?. 
Buccin trombone. Nominal pitch: 8?. 
Technical description: Brass; ligature fitting on bell section at joint; stockings 
on main slides. Bell with one coil, angled to face forwards. Repair History: 
Main slide possibly not original (tenon of slide section of joint is tapered, bell 
section joint for cylindrical tenon). 
Type: Buccin 
subject: Musical Instruments 
Date: Circa 1840 
Place/Time: Probably France 
Institution: University of Edinburgh 
Provider: MIMO - Musical Instrument Museums Online 
Providing Country: Europe 
First Pubfished In Europeana: 2013-11-22 
Last Updated In Europeana: 2013-11-22 
Dataset: 09102 Ag_EU MIMO ESE 
Place/Time: Probably France 
Buccin trombone. Nominal 
pitch: 8?. 
Buccin trombone (bell 
only). Nominal pitch: 
OUR SITES 
LANGUAGE 
p 
± DOWNLOAD 
FIND OUT MORE 
View at University Of Edinburgh e. 
CAN 1 USE IT? 
Limited Re-use 
€@SO cc BY-NC-SA e 
SHARE 
00000 
E? piccolo. Nominal pitch: 
D?. and 
show more... (80,811) 
SUBJECTS, RESOURCE TYPES, GENRES AND FORMS (CONCEPTS) 
We transform the world with culture! We want 
to build on Europe's rich heritage and make it 
easier for people to use, whether for work, for 
learning or just for fun. 
MORE INFORMATION 
About 
Development updates 
All institutions 
Become our paltner 
Contact us 
HELP 
Search tips 
Terms of use 
& Policies 
Co-financed by the European union 
Connecting Europe Facility 
SIGN UP FOR OUR NEWSLETTER 
EMAIL ADDRESS 
@ English French 
FIND US ELSEWHERE: 
0000 

Figure 3 example of a record mapped to EDM classes

## Research on Annotations

In order to fulfill the requirements, another research on annotations was necessary to conduct.

While working on the research, we have witnessed history happening as we were browsing the web and documentations, especially under W3C web site. 3 recommendations were published on W3C’s web site which set the standards of annotations and opened new horizons for the web. (W3C.org annotations announcement, n.d.):

**Web Annotation Data Model**: the standard offers a structured model and format, in JSON, for annotations to be interoperable across platforms. Data Model consists of 3 main parts: Body and Target being related and Annotation pointing to both on a top layer. In the below JSON, the basic terms can easily be observed: (Annotation Data Model, n.d.)

{

"@context": "http://www.w3.org/ns/anno.jsonld",

"id": "http://example.org/anno38",

"type": "Annotation",

"motivation": "commenting",

"creator": {

"id": "http://example.org/user1",

"type": "Person",

"name": "A. Person",

"nickname": "user1"

},

"created": "2015-10-13T13:00:00Z",

"generator": {

"id": "http://example.org/client1",

"type": "Software",

"name": "Code v2.1",

"homepage": "http://example.org/homepage1"

},

"generated": "2015-10-14T15:13:28Z",

"stylesheet": {

"id": "http://example.org/stylesheet1",

"type": "CssStylesheet"

},

"body": [

{

"type": "TextualBody",

"purpose": "tagging",

"value": "love"

},

{

"type": "Choice",

"items": [

{

"type": "TextualBody",

"purpose": "describing",

"value": "I really love this particular bit of text in this XML. No really.",

"format": "text/plain",

"language": "en",

"creator": "http://example.org/user1"

},

{

"type": "SpecificResource",

"purpose": "describing",

"source": {

"id": "http://example.org/comment1",

"type": "Audio",

"format": "audio/mpeg",

"language": "de",

"creator": {

"id": "http://example.org/user2",

"type": "Person"

}

}

}

]

}

],

"target": {

"type": "SpecificResource",

"styleClass": "mystyle",

"source": "http://example.com/document1",

"state": [

{

"type": "HttpRequestState",

"value": "Accept: application/xml",

"refinedBy": {

"type": "TimeState",

"sourceDate": "2015-09-25T12:00:00Z"

}

}

],

"selector": {

"type": "FragmentSelector",

"value": "xpointer(/doc/body/section[2]/para[1])",

"refinedBy": {

"type": "TextPositionSelector",

"start": 6,

"end": 27

}

}

}

}

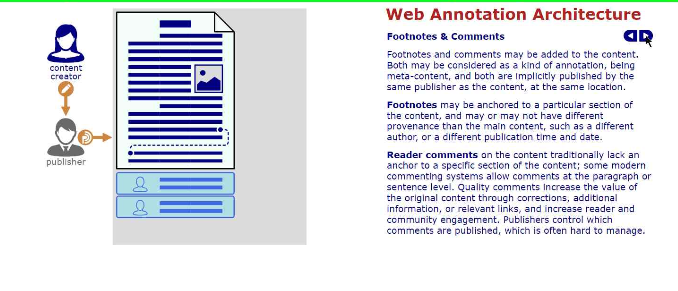
**Web Annotation Vocabulary**: provides the definitions to all entities to be used in Web Annotation Data Model. The standard specifies the RDF classes and provides the JSON-LD Context and profile definitions needed to use the Web Annotation JSON serialization in a Linked Data context. JSON-LD context and the full list can be referenced from <https://www.w3.org/ns/anno.jsonld>.

**Web Annotation Protocol**: describes the transport mechanisms for creating and managing annotations which is same as the methods used in Web Architecture and REST. There standards added to these methods are:

* The media type to use for Annotations is: application/ld+json;profile="http://www.w3.org/ns/anno.jsonld
* Annotation Containers are constrained by the set of constraints described in this specification, and thus the ldp:constrainedBy URL is http://www.w3.org/TR/annotation-protocol/
* The link header can refer from any resource to an Annotation Container using a rel type of: http://www.w3.org/ns/oa#annotationService
* The response from a Container after creating an Annotation should include a representation of the Annotation, after any changes have been made to it, in the JSON-LD serialization.
* Annotation Containers should only contain Annotations, and not other resources. (W3.org Annotation Protocol, n.d.)

As recommended by the W3C, the workflow of the annotations can be followed easily through the diagram they have published as part of the announcement. (w3.org annotations working group, n.d.)

Below you could find the visual guideline.



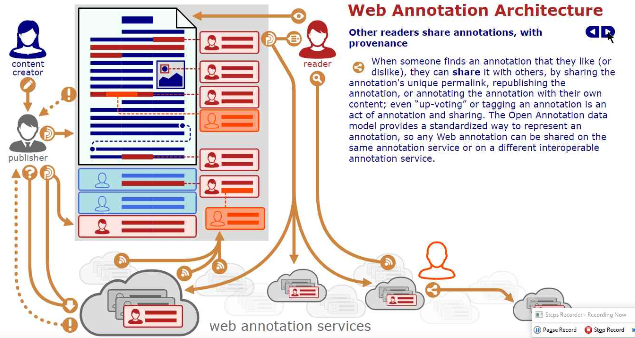
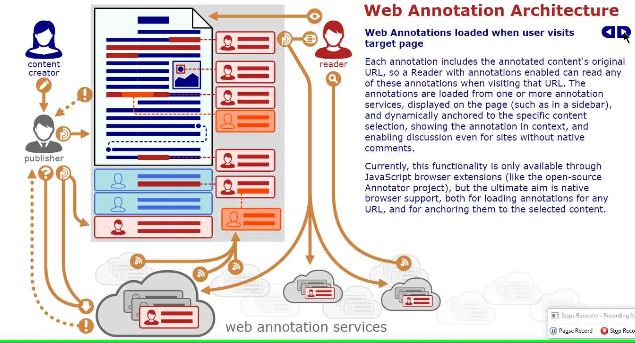
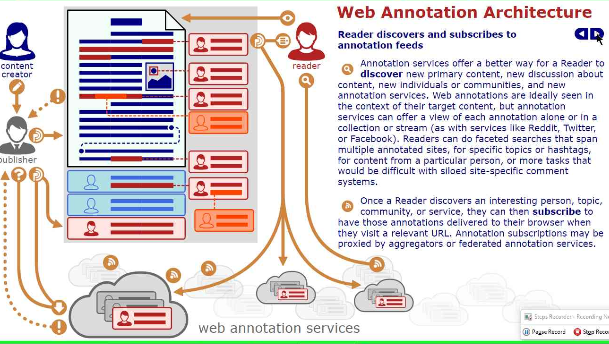
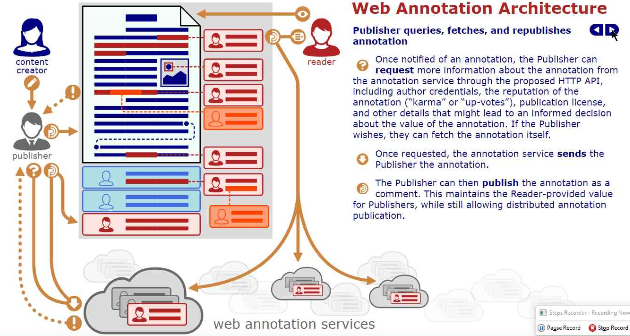
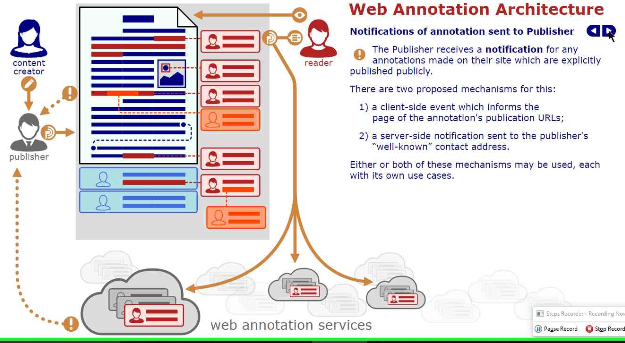
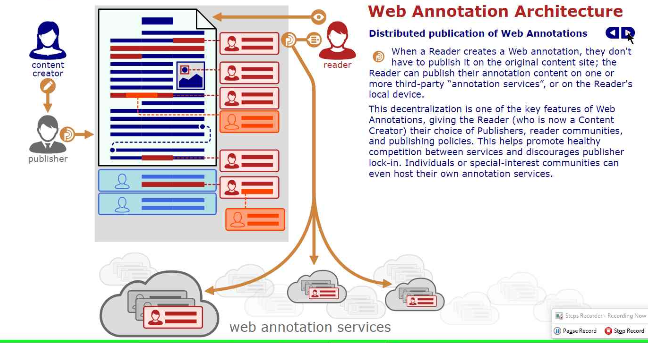
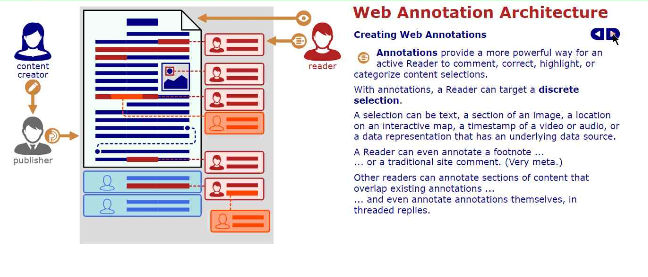


Figure 4 Web Annotation Architecture Explained (Visual Guideline)

The W3C standards enable users to comment on any web site or content they want without the need of controlling authority. It just adds another layer or a veil as to speak, to the WWW as we know it. This feature most probably be available as a native feature in the browsers since it has become a standard.

The W3C architecture provides a model where annotations are stored separately on another layer on web sites and reappears in real-time whenever the relevant web page is viewed. The control shifted from the publisher to the user. Web annotations will enable users and communities to use annotations across any internet-connected document, whether in HTML, PDF, EPUB, or other formats. Although there are many details to figure out, standards are a good start. (Hypothes.is Blog, n.d.)

The W3C standards enable users to comment on any web site or content they want without the need of controlling authority. It just adds another layer or a veil as to speak, to the WWW as we know it. This feature most probably be available as a native feature in the browsers since it has become a standard.

The W3C architecture provides a model where annotations are stored separately on another layer on web sites and reappears in real-time whenever the relevant web page is viewed. The control shifted from the publisher to the user. Web annotations will enable users and communities to use annotations across any internet-connected document, whether in HTML, PDF, EPUB, or other formats. Although there are many details to figure out, standards are a good start. (Hypothes.is Blog, n.d.)

## Hypothes.is utilized as annotation benchmark

The Hypothesis is a [single-page web application](https://en.wikipedia.org/wiki/Single-page_application), which can be called embedded as an internal solution to the Content Management System. **Hypothesis is totally in line with the latest W3C standards accepted on annotations.** It has ist standards and principles set up just like the recommended standards. Hypothesis is:

* Free, open source software using open standards
* Works across platforms and formats
* Non-profit
* Neutral
* 100% community moderated
* Merit based
* Pseudonymous
* International
* Transparent, auditable approach in systems and in governance.
* Many formats, many contexts
* HTML, PDF, video, books. News, blogs, scientific articles, legislation, regulations, Terms of Service, etc.

As the latest web annotation standards suggest, it adds an annotation layer to any content an annotation service runs on a remote server.

Data can flow in both directions: from the annotated content to the client and vice versa, just like the annotation data model suggest the relationship between source, target and the annotation. Communication with the annotation service is also bidirectional, making use of an HTTP API and a WebSocket connection.

User 
I Annotated 
I content 
> I Annotation 
Client 
service 

There are two crucial parts to the system:

* 1. Between the client code, executing in a browser, and the service, executing on a remote server.
  2. Between the annotated content (which may be an HTML page or a PDF rendered as an HTML page) and the client application. This boundary is marked with asterisks (\*) in the ASCII art above.

Hypothesis utilizing the web annotation data model, web annotation vocabulary and web annotation protocol. Especially the latter one is highly important focusing on the transport mechanisms for creating and managing annotations which is same as the methods used in Web Architecture and REST. Moreover, it focuses on trust and security during the transport of data by providing:

* user credentials
  1. annotation data or metadata which is displayed by the client
  2. user profile information
  3. group membership records
  4. user search history

**Same-Origin Policy Protections:**

[Same-origin policy](https://en.wikipedia.org/wiki/Same-origin_policy) (SOP) is the base for many client-side security mechanism, which means content residing on one place has limited control over content on another place.

the Hypothesis client application executes within an <iframe> inserted into the annotated content just like it is structed in SOP.

This <iframe> which is the annotation in web annotation data model, has an origin distinct from that of the hosting page and SOP rules apply. The components of the client which execute in the annotated page must communicate with the client frame using [cross-document messaging](https://en.wikipedia.org/wiki/Web_Messaging). (Hypothes.is document on security, n.d.)

We have also made use of the Hypothesis API in the back end, mobile and web development. We have simply used the REST structure which crucial part of the web annotation data model and API is in line with the standard. More examples can be found under <https://hypothes.is/api/annotations>. An example of the post annotation template can be viewed below: (Hypothes.is API creating annotations, n.d.)

{

"group": "string",

"permissions": {},

"references": [

"string"

],

"tags": [

"string"

],

"target": [

{

"selector": [

{

"type": "string"

}

]

}

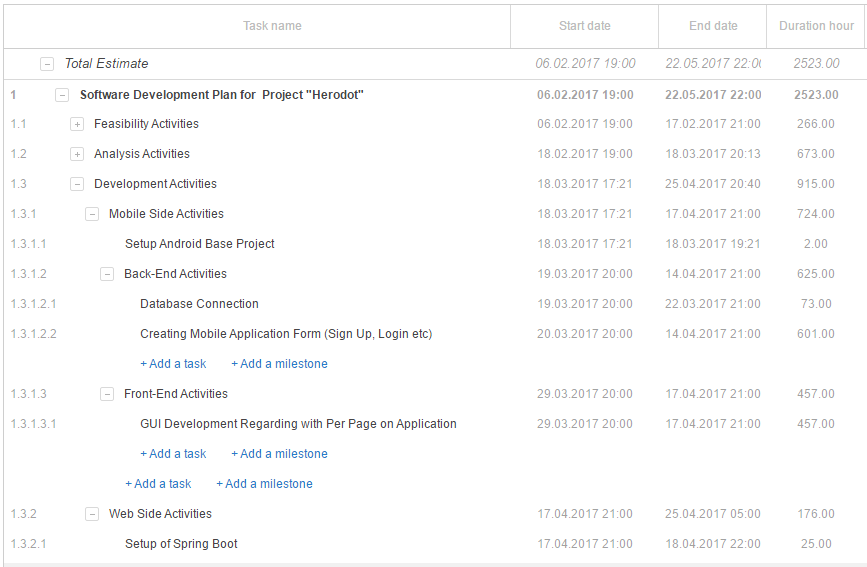
],

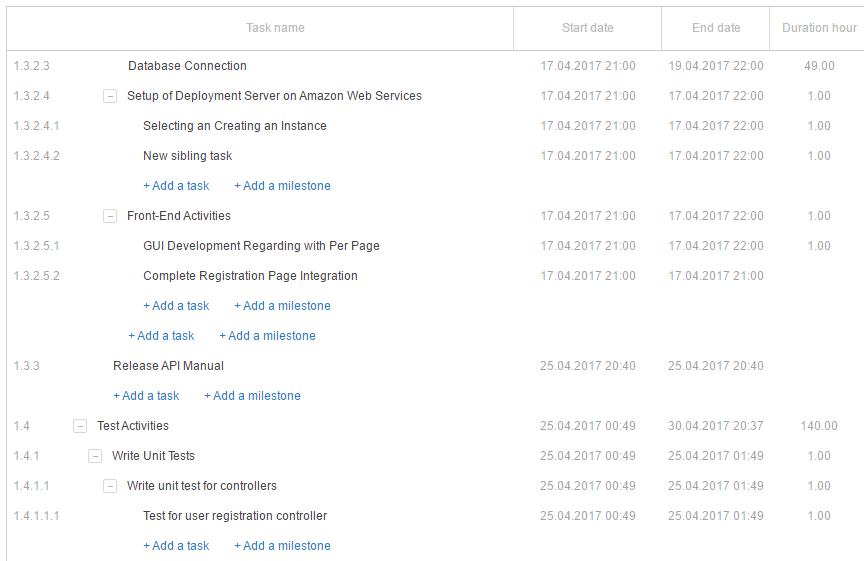
"text": "string",

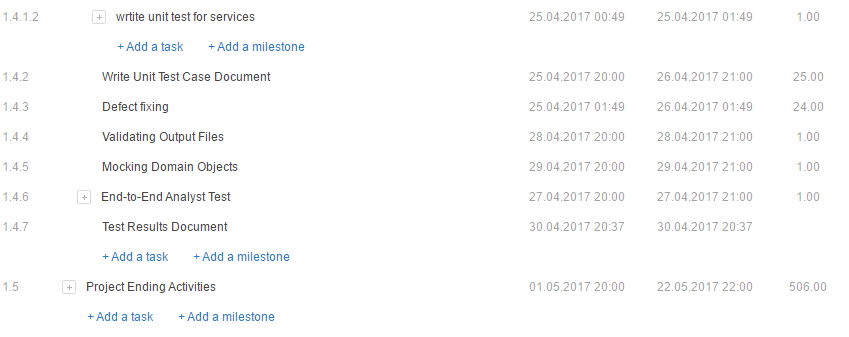
"uri": "string"

}

# Project Plan







# Milestone Reports

## Milestone Report #1: Project Management Decisions (Closed on 20th of Feb)

This “Project Management Decisions” milestone covers the two goals – as a questionnaire - stated below

1. “How should we handle project planning?”
2. “Decisions on project and development management”

As a team, we use Github’s issue feature as the technical communication tool to assign and track tasks. The initial question has been lurked in order to provide a planning report for the end of this semester. Individual and team tracking tools are discussed over the issue opened on Github.

As a result we have decided to track project and development management under the Wiki tab on Github.

**The “Project Management” part consists the sections as it follows;**

1. Sprints – We use Scrum as the project management methodology. We briefly explained the road map for retrospective, sprint planning and sprint review phases on the associated Wiki page.
2. Board Management – This section describes the scrum board (To-Do, In Progress and Done sections) and how do we select tasks in order to deliver them.
3. Issue Management – Issue description and appropriate labelling.

**The “Development Management” part consists of;**

1. Modules – API, front-end and mobile. Each module has been represented on the individual folder on the repository.
2. Branching – Git flow will be followed. Each commit should start with the associated issue number. Developer communication is necessary when there are conflicts on the issue where which developer is responsible.
3. Testing – Every developer should test their code before sending it to review.
4. Unit Tests – They will be practiced after the code is written.

As a team, we collect the meeting notes on the “Group Communication” section under the Wiki index. We also shared a contributing guideline page to create knowledge in the group. The contributing guideline documentation has been completed as the second milestone on the sprint 2. “Contributing Guideline” could also be found under the Wiki index.

As a result, the project management road map stated above describes our working agreement for the “Heredot” project. We achieved to find a balance to track tasks and maintain the work flow over the semester. This part has been all up voted and accepted by the entire team so we closed the issue on the 20th of Feb. We also closed the milestone successfully on 20th of Feb.

## Milestone Report #2: Contributing Guidelines Documentation (Closed on 4th of March)

This milestone consists of the issues – successfully closed - stated below:

1. “Creating a basis project in Android”
2. “Creating a Contributing Guideline Documentation”
3. Naming the project

* Android Studio is used in order to create new Android project with setup environment and it’s been also connected to Git to push the project under the “ANDROID” folder.
* As a team we wanted to create a Wiki page to guide the developer through the contribution phase. The page consists of the information to creating a local clone of one’s fork, syncing one’s fork with the actual Herodot repository, keeping one’s local fork updated, making changes and creating pull requests.
* This procedure document gives insight to the developer to tell how to integrate his/her work with Git. This guideline also aimed to create knowledge in the development team. Git branching model, Git cheat sheet and official documentation referenced to create it.
* The debate is held on to decide what to call our project is resulted as “Herodot” with everyone’s up vote.
* The main aim in this milestone was creating guideline documentation for contribution as a developer. However, we also achieved to provide a new project name and creating a base project in Android as side additions.

## Milestone Report #3: Analysis Document (Closed on 8th of March)

This milestone consists of the issues – successfully closed - stated below. The main aim in this milestone was about to provide an extended analysis document that is including mock-up design with realistic data, updated requirements elicitation and insight of coding.

1. “Icon for Herodot”
2. “Mock Up Interface Design for Website”
3. “Deciding on Cultural Heritage Items and Attributes”
4. “Hypothesis and Europeana Hands on Experience to Work on Requirements Elicitation and Mock Up Details”
5. “Web Annotation Data Model Research”
6. “Mock Up Design with Regards to Interface Design for Mobile App”
7. “Requirements Elicitation”

As the design concept we followed to create mock-ups for both mobile and web ends. We decided to use this mobile presentation tool named InvisionApp to present our mock up work. The main dilemma was to refine the requirements properly in order to create a proper mock-up design where all the features are stated correctly and satisfying. We also opened an issue to discuss and find a midpoint to decide our project’s icon.

We finalized the icon with the most voted one.

We created this issue named “Requirements Elicitation” to discuss and to come up with the questions first. Once the online requirements document has been uploaded we all gathered around with new ideas and added them orderly.

We have prepared some questions to throw ideas and knit-picking process in class. The answers were really helpful for us to have a concrete idea about the application. Ms. Uskudarli has explained what is required from the project from a customer’s eye.

We decided to go on and research Web Annotation Data Model and came up with the idea of limit our scope to not to over engineer. We have found multiple – over 15 – sources to gather information including Web Annotation Protocol Server Implementation with Java Spring. After sharing lots of research links we have decided to create a “Research” section as a part of project’s index and started to collect them as a consolidated research. The draft of the research on web annotation is done by using a web annotation tool and saved under an online CMS system to have a prior practice to be wikified with the findings agreed as a team.

Hypothesis and Europeana sources are shared to be investigated for benchmarking on an issue. A profile and a group on cultural heritage is created on hypothesis, so group members could have hands on experience and benchmark with existing solutions. As a result of the mock up design phase both heritage items and attributes are investigated to be included in the document. The research done on determining cultural heritage categories and properties are compiled under the associated wiki page.

We have presented the whole document and designs on the class. The milestone successfully closed on the 8th of March in order to move on to the next work package.

## Milestone Report #4: Project Planning And High Level Architecture Formation (Closed on 3rd of Apr)

This milestone consists of the issues stated below:

1. High Level Design
2. Create Base Project in Spring Framework
3. Create Base UI Project
4. Project Plan Design

A brief summary about the milestone could be also found below:

High level architecture and design are created for the project. And the base UI and application is created for the web.

1. High Level Design

High level UI/UX design is done. Draw.io tool is used for design the page as mock-up. Some design issues are discussed as a part of that.

2. Create Base Project in Spring Framework

Base project is created and submitted to the repository. Spring Framework is used for the project. This part will form the back-end side of the project.

3. Create Base UI Project

Base UI project is created based on high level design decisions. HTML, CSS and JS is used for web design. Angular.js 2 is used for front-end part of the project. Base project supports general structure of Angular.js 2 and has basic authentication and navigation features.

4. Project Plan Design

Firstly, big blocks for the project are decided. And then these big blocks divided into smaller parts and details of the project plan is designed. GanttPro tool is used for design the project with gantt chart. Milestones and sprints are decided and clarified in detailed.

Sign in credentials:

e-mail: taygun.gokdemir@hotmail.com

password: Herodot2017

GanttPro Link: https://app.ganttpro.com/#!/app/home

All the issues has been commented on and discussed on Github as a team.

## Milestone Report #5: Sanity Check with Registration Prototype (Closed on 10th of April)

This milestone consists of the issues stated below:

1. Mobile Design & Development
2. Milestone integration to the project plan
3. Web UI improvements on Add Heritage Item

A brief summary about the milestone could be also found below:

We have started implementation of backend, web front-end and mobile components individually. These components come together and communicate without any issues. Registration functionality is a good candidate for this sanity check as it is one of our requirements. We will need to have the backend deployed so that our front end components can consume the API it provides.

1. Mobile Design & Development

This issue includes the first version of the mobile application design. This design work consists a launch page and registration (signup and login) - as device can keep credentials the user shall not be expected to login every time user opens the app. We have completed the first version and presented on the class on 10th of April.

We have received a positive feedback from Ms. Uskudarli.

2. Milestone Integration to Project Plan

After our presentation session for the project plan milestone, Ms. Uskudarli has suggested us to collect milestone reports as inventory for the associated milestones. She shared a requirement list for the milestones on Piazza as well.

We have started to keep milestone reports from then and present them in class as a ritual. Our milestone reports cover about the issues that we closed, our aim for that sprint and what is our next step.

We have also integrated the missing milestones to our project plan as well. We finalized the complete milestone list on the 3rd of April on our team meeting.

3. Web UI Improvements on Add Heritage Item

The improvement progress for the "Add Heritage Item" consisted below work:

* Changes made based on web mock-up v3
* Adding Heritage Item page is redesigned
* Logo is added

All the issues have been commented on and discussed on Github as a team.

## Milestone Report #6: Create System V1 (Closed on 17th of April)

This milestone covers the issues stated below:

1. Placeholder image for each CHO category
2. Annotator.js integration and showcase

A brief summary about the milestone could be found below:

We have integrated the back-end and front-end and annotation level to run the first tests of version 1 and to continue with iterative versions of the system.

1. Placeholder image for each CHO category

We have created thumbnail picture for each category in CHO in line with bootstrap specs and image resolutions.

2. Annotator.js integration and showcase

We have completed the steps of loading annotation library and creating a simple page to showcase API.Edited with project pipeline, assignee, milestone, estimate to show up on project board with valid data.

It was important for us to do some research on Cultural Heritage Object categories to integrate our project since users need to filter their search. So a user will be able to start to limit his search criteria.

## Milestone Report #7: View System V2 (Closed on 20th of April)

This milestone consists of the issues stated below:

1. Login and Sign-up Page Updates
2. Create add cho item page with new template

Basic summary on the milestone:

Database should be configured and created components should be set out image annotation and location annotation should be integrated mobile and front end integration with back end should iterate to v2 complete user workflow with at least 1 possible scenario should be offered by the system and this flow should be recorded in the system with all data integrations.

1. Login and Sign-up Page Updates

Form structure is created on Login and Sign-up pages to provide a proper design for registration. New logo also has been added.

2. Create add cho item page with new template

We have completed CHO research to create knowledge. After reading the Europeana Data Model, and linked data and its workings, the logic and technology behind semantic web and annotations and their data models, in addition with standards, it was evident that a schema was already in place and as software engineers we needed to understand and use these standards in order to deliver the requirements asked by the client.

The client would be happy to know that the end product/ web site with its data and metadata will be interoperable in web and may also have a value proposition to be more visible than other resources by the use of these linked in data benefits and be a pioneer in the new linked in data network in progress. More inbound means better site score, data is more enriched and visible. Trustworthy linked open data being made available now, gives you big advantage in the www in terms of users, traffic, etc etc. by being a canonical reference point.

In short after reading all the standards in place on the linked data structure, it was meaningless to come up with a new data properties. As software engineers we will be using already established standards in our domain according to our needs.

We have provided detailed analysis report on our wiki page called "Research on Cultural Heritage" to collect knowledge. This milestone especially helped us to figure out the category - tangible vs. intangible - of heritage items.

## Milestone Report #8: Edit System V3 (Closed on 27th of April)

The milestone consists of below issues:

1. Maps Views to Add CHO Item Page1
2. Annotator.js Integration and Showcase
3. Herodot API Deployed to api.herodot.world

**General Thesis on the Milestone**

Model should be working on a stable phase as well as the DB. CHO should be working like a CMS on its on CRUD should be consumed.

Adding maps views one of the goals to be completed within this milestone. So this way, multiple time location pairs could be added by a user for same CHO item.

Name Time (with Resolution Selector) Location (which should be selected on map)

Annotator.js integration included below tasks;

* Loading annotation library
* Creating a Simple page to showcase API on class

The above matters eventually edited with project pipeline assignee, milestone, estimate to show up on project board with valid data. Although We have faced with some issues regarding this. Hypothesis wasn't backwards compatible to annotator.js 1.2.x.

Registration API endpoint completed.

## Milestone Report #9: Iteration V4 (Closed on 20th of May)

The milestone consists of below issues:

1- Connect Front End to API

The API is documented. API documentation can be accessed via this link:<https://github.com/AyranIsTheNewRaki/Herodot/wiki/API-Documentation>

API is deployed to this link:

<http://api.herodot.world/>

API is connected to the Front End code in this sprint.

## Milestone Report #10: Iteration V5 (Closed on 27th of May)

This milestone consists of the issues stated below:

1. CRUD Implementations for CHO items on web

2. List CHO Items on Mobile Application

3. Template Folder Update

Basic summary on the milestone:

This milestone mostly contains development of new features for the applications. Development for both mobile and web applications are included in the milestone.

1. CRUD Implementations for CHO items on web

CRUD implementation done for CHO items on web application. Front-end and API had connected previous sprint. Now, CRUD implementation is enabled for the users.

2. List CHO Items on Mobile Application

CHO items are listed for logged in users. User are able to see CHO items in the system. Image, time, location etc. attributes of the item became viewable. Adding annotation activity and some other CRUD implementations for CHO items are leaved for later period.

3. Template Folder Update

Index, Add Item and Item pages are updated. Navbar, Images and annotation.js is added to the project. Category CHO pages created.

## Milestone Report #11: Iteration V6 (Closed On 29th Of May)

This milestone consists of the issues stated below:

1. Create Add-List CHO Pages on Web Application

2. Homepage Design

3. Add Category Images

4. Add Annotations to CHO on Mobile App

Basic summary on the milestone:

This milestone includes final tasks for the project. Mostly implementation of annotations took time. Also some nice to have tasks like homepage design and adding category images are done in this sprint.

1. Create Add List CHO Pages on Web Application

Add CHO item page finished. Integration with Google Maps and Herodot API is done. Listing CHO items is completed.

2. Homepage Design

Homepage is designed for web application.

3. Add Category Images

Category images are uploaded to the system.

4. Add Annotations to CHO on Mobile App

Annotation with Hypothesis are added to the mobile application. Annotations are being sent both to Hypothesis and our Backend. For each CHO annotation id's are kept in the backend and details are fetched from Hypothesis.

# Requirements Elicitation

Below the reader could find requirements elicitation regarding discussions throughout the semester.

1. The system shall have a mobile application and a web application
2. There shall be 6 pages on mobile and web
   1. Homepage
   2. Category Items
   3. Heritage Item Detail
   4. Sign Up and Login
   5. Add Heritage Item
   6. My Heritages
3. Annotations shall be only available on Heritage Item Detail pages, which contain detailed information about specific item
4. Annotations shall be only in text and image format
5. Annotations and heritage items shall be separated
6. Annotations shall be able to written by anyone on the item heritage page
7. There will be two user types
   1. Visitor
   2. Member
8. Members shall be able to do everything that visitors can do
9. Visitors shall browse Homepage, Heritage Item Detail pages, Sign Up page and Login page.
10. Visitors shall see annotations on Heritage Item Detail pages.
11. Visitors shall see all annotations on the page if they don’t select any part of website
12. Visitors shall see only related annotations when they select a part of a website
13. Visitors shall be able to see the part of web page where an annotation refers
14. Visitors shall be able to sign up.
15. Visitors shall be signed up with their name, e-mail address and password
16. Visitors shall be turn into members when they are signed up to the system
17. Visitors shall be able to login to the system if they already signed up to the system
18. Visitors turned into Members when they logged in to the system
19. Members shall be able to add heritage items
    1. Members shall be able to add heritage items with map featurette
20. Members shall be able to browse heritage items that they added on My Heritages page
21. Different members shall be able to add same heritage items.
22. Members shall be able to take instant photo when add a heritage item on mobile application
23. Heritage items shall have item types
24. Each heritage object shall have the following set of properties:
    1. Members shall be able to add description to the item in free text format
    2. Members shall be able to add format of the actual cultural heritage item like paper, oral, stone etc.
    3. The unique identifier shall be provided by the service provider in case of a new instance creation like: RP-T-1952-380
    4. Members shall be able to add language in the ISO 639 2 0r 3 letter primary language tag like: tr, se, ar
    5. Members shall be able to add subject of the CHO like: trombone, statue etc.
    6. Members shall be able to add the current location of the CHO with coordinates or city name
    7. Members shall be able to choose the item's related category from the categories provided in the web site schema and database item type like: tradition, architecture.
    8. Members shall be able to add the database type of the item like: image or text.
    9. Members shall be able to add construction beginning date to the item
    10. Members shall be able to add end construction date to the item

# Software Design Document

## High Level Entity Relationship Diagram

Below the reader could find the high level E-R Diagram corresponding to the solution.

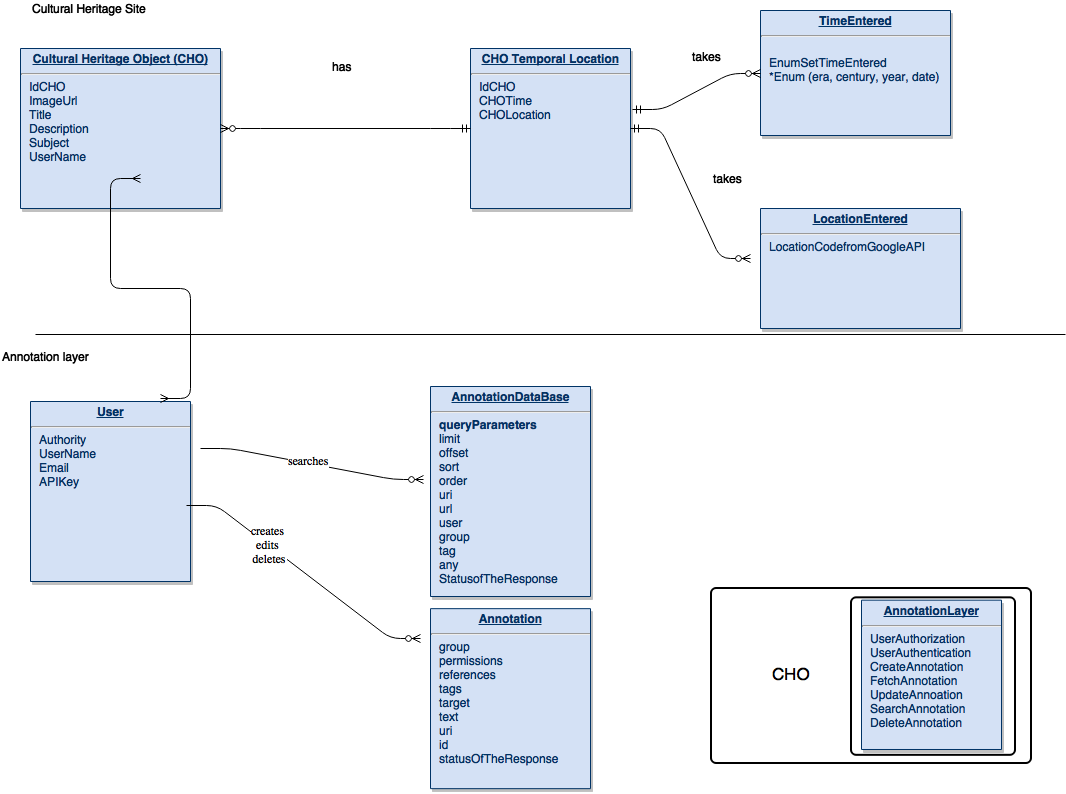


Figure 5: High level ERD

## System Component Diagram for Web

Below the reader could find the system component diagram for web.

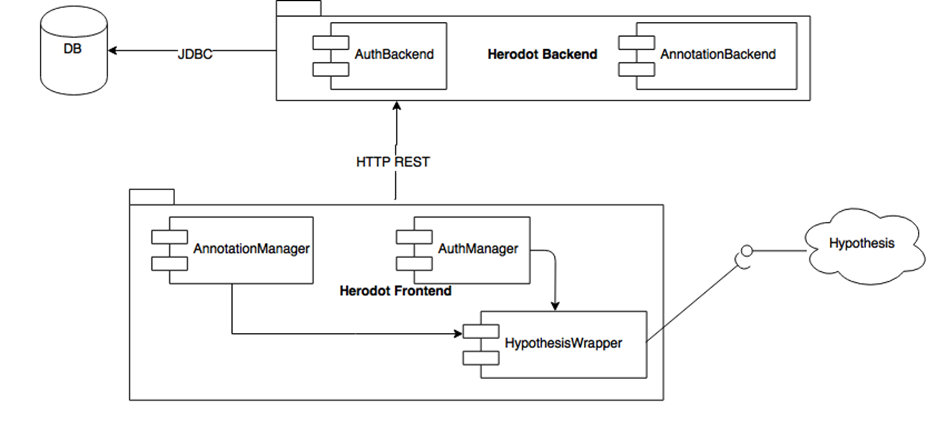


Figure 6: System Component Diagram

## System Component Diagram for Mobile

Below the reader could find the system component diagram for mobile.

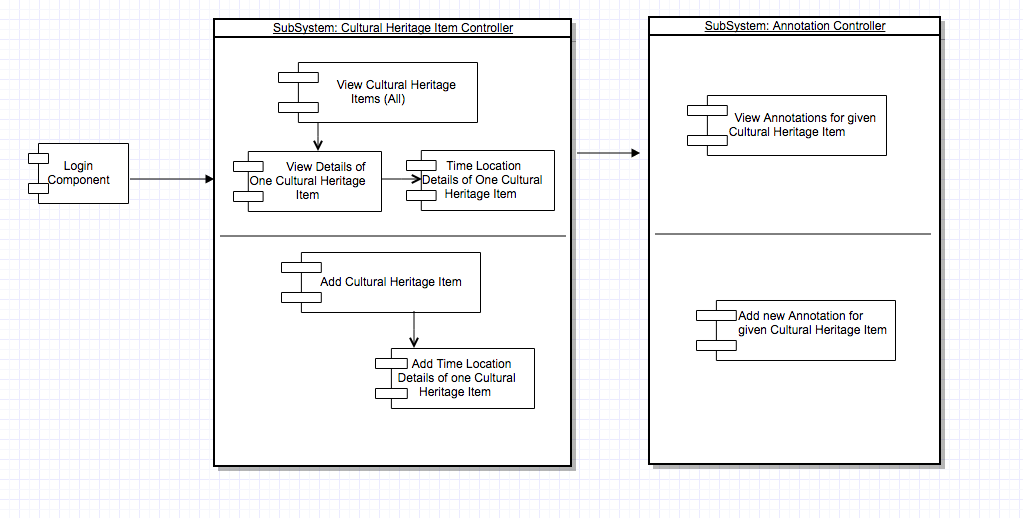


Figure 7 Mobile System Component Diagram

## System Sequence Diagram for Web

Below the reader could find the system sequence diagram for web side.

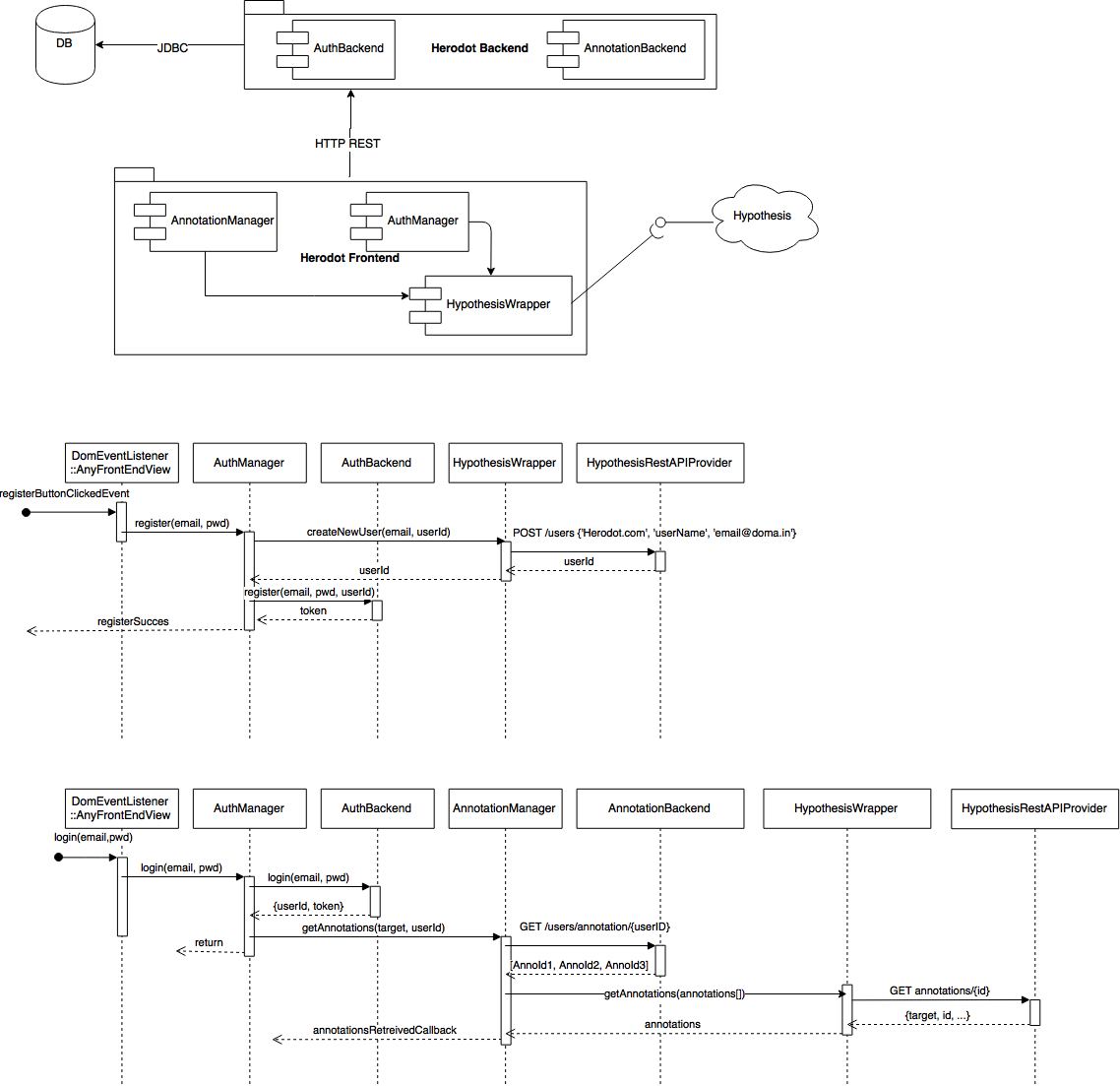


Figure 8: System Sequence Diagrams for Heredot

## Mobile Sequence Diagram

Below the reader could find the sequence diagram for mobile side.

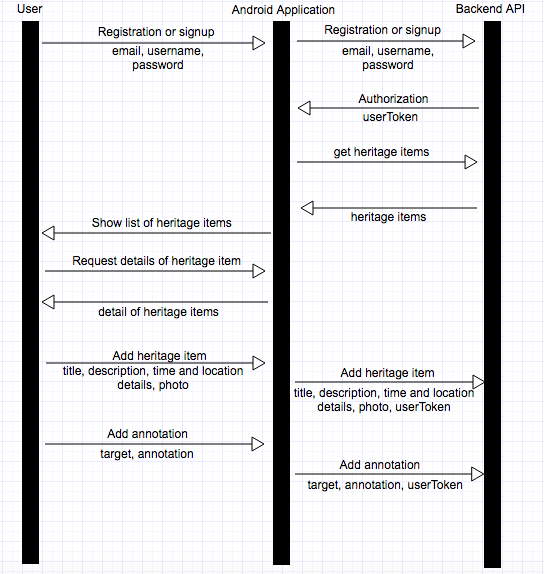


Figure 9: Sequence Diagram for the Mobile

## System Deployment Diagram

Below the reader could find the system deployment diagram.

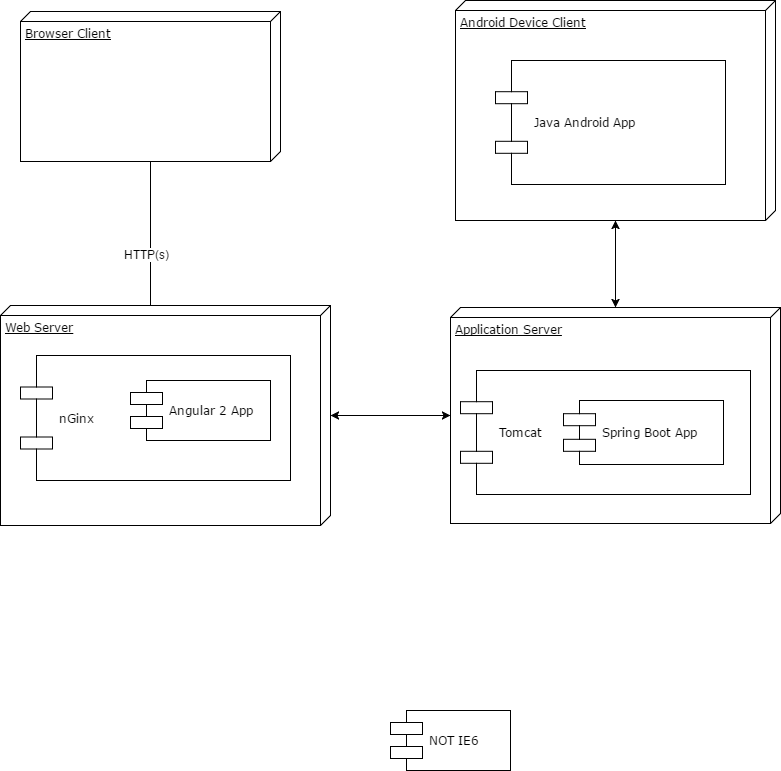


Figure 10 System Deployment Diagram

## Mock-up Diagrams

Below the reader could find mock-up diagrams regarding web and mobile side.

### Web Side Mock-Ups

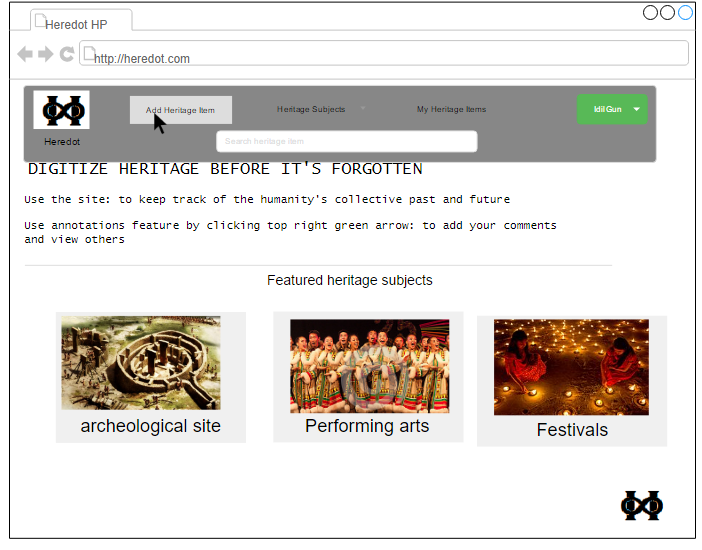


Figure 11 Heredot Mockup HomePage



Figure 12 Heredot Mockup Add CHO page

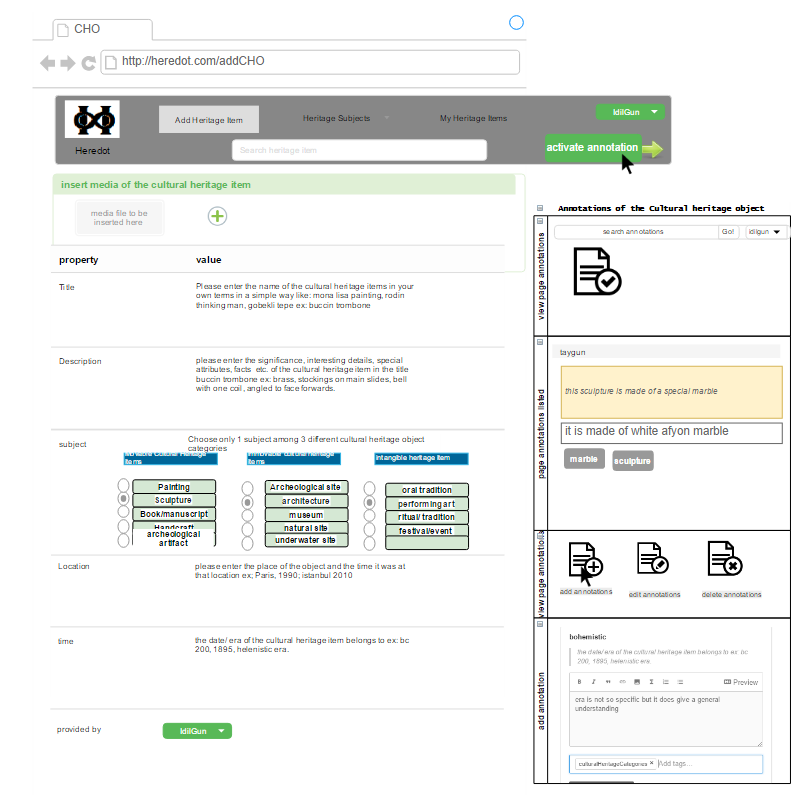


Figure 13 Heredot Mockup Add Annotations

### Mobile Side Mock-Ups

Below the reader could find mobile side mock-up works.

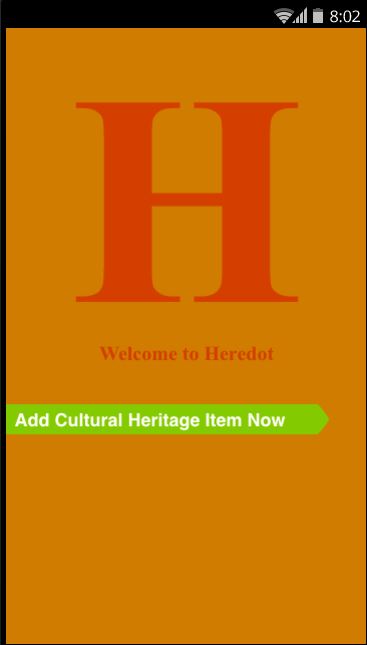


Figure 14 Mobile Home Screen Mock Up



Figure 15 Mobile Mock Up for Categories Page 1

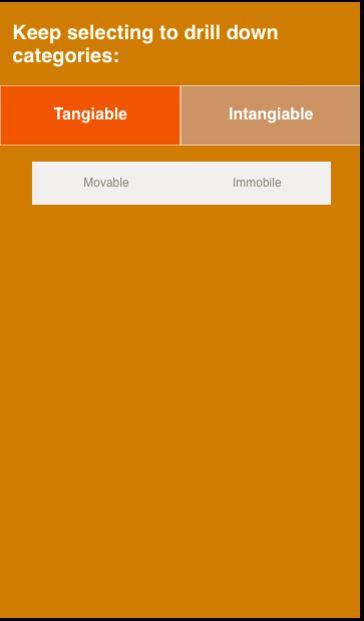


Figure 16 Mobile Mock Up for Categories Page 2

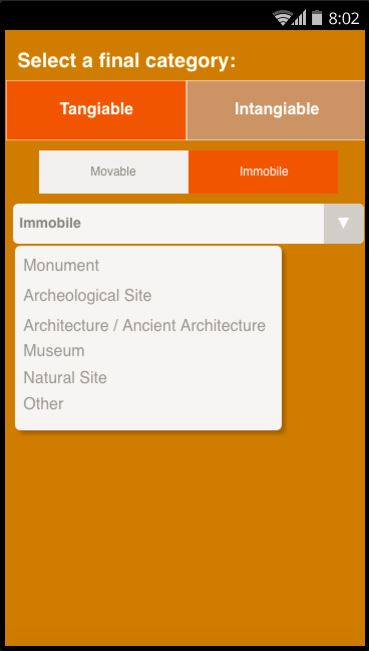


Figure 17 Mobile Mock Up for Final Categories Page

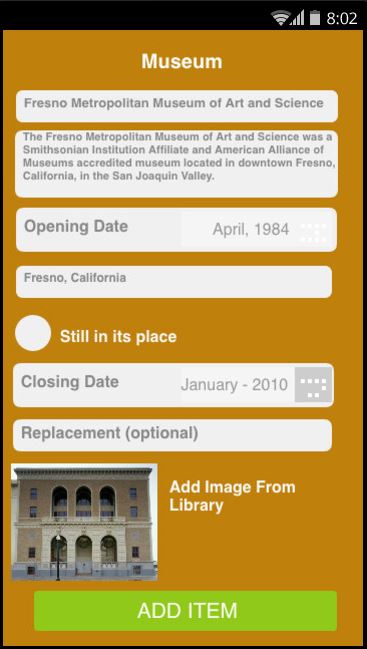


Figure 18 Mobile Mock Up for Heritage Item Page

# System Specifications Document

## System Specifications Summary

### Mobile (Android):

The Android application "Herodot" is developed using Android Studio 2.3.1

Java Version: 1.8.0\_112

JVM: OpenJDK 64-Bit

Gradle is used to handle project dependencies.

Android Package Name (application id): com.ayranisthenewraki.heredot.herdot

Compile SDK Version / Target SDK Version: 24

Min SDK Version: 16

Build Tools Version: 25.0.2

Most of the user testing is done on Nexus 5 API 25 (Android 7.1, API 25)

### Front-End:

Bootstrap 3.3.7 utilizing CSS, LESS

Angular libraries used:

• "@angular/common": "~2.4.0" contains multiple interfaces that are required for all angular applications

• "@angular/compiler": "~2.4.0" contains the html parser/compiler that is required to use angular templating. This can be removed if some other templating strategy is used.

• "@angular/core": "~2.4.0" contains the core framework definitions,

• "@angular/forms": "~2.4.0" contains libraries that are required for using forms,

• "@angular/http": "~2.4.0" contains interfaces for xmlHttprequests and async opreations

• "@angular/platform-browser": "~2.4.0" contains interfaces that bootstraps an angular application for the chosen browser specification. For example user can choose to support es6 only browsers.

• "@angular/platform-browser-dynamic": "~2.4.0" bootstraps angular application for any browser independent of the standards it supports.

• "@angular/router": "~3.4.0" contains interfaces for navigation between components. Historically there have been community implementations of the router.

• "core-js": "^2.4.1" contains polyfills that enable web standard compatibility.

• "rxjs": "5.0.1" contains interfaces for using many data structures asynchronously. For example user can create a stream from an array similar to Python’s list comprehensions.

• "systemjs": "0.19.40" browser independent module loading library

• "zone.js": "^0.7.4" contains interfaces to create scopes for asynchronous operations. Angular libraries used:

• "@angular/common": "~2.4.0" contains multiple interfaces that are required for all angular applications

• "@angular/compiler": "~2.4.0" contains the html parser/compiler that is required to use angular templating. This can be removed if some other templating strategy is used.

• "@angular/core": "~2.4.0" contains the core framework definitions,

• "@angular/forms": "~2.4.0" contains libraries that are required for using forms,

• "@angular/http": "~2.4.0" contains interfaces for xmlHttprequests and async opreations

• "@angular/platform-browser": "~2.4.0" contains interfaces that bootstraps an angular application for the chosen browser specification. For example user can choose to support es6 only browsers.

• "@angular/platform-browser-dynamic": "~2.4.0" bootstraps angular application for any browser independent of the standards it supports.

• "@angular/router": "~3.4.0" contains interfaces for navigation between components. Historically there have been community implementations of the router.

• "core-js": "^2.4.1" contains polyfills that enable web standard compatibility.

• "rxjs": "5.0.1" contains interfaces for using many data structures asynchronously. For example user can create a stream from an array similar to Python’s list comprehensions.

• "systemjs": "0.19.40" browser independent module loading library

• "zone.js": "^0.7.4" contains interfaces to create scopes for asynchronous operations.

### Back-end:

Web application "Herodot" is developed using Java 8 and Spring Boot

Java Version: 1.8.0\_112

JVM: OpenJDK 64-Bit

Maven is used to handle project dependencies.

Web Package Name (application id): world.herodot.backend

Compile SDK Version / Target SDK Version: 24

Min SDK Version: 16

Build Tools Version: 25.0.2

## System Specifications Details

**Heroku** supports deployments with git commands.

All you need to do is to create a git repository for the project by typing "heroku create herodotbackend"

The "heroku create" command creates a new application on Heroku – along with a git remote that must be used to receive your application source.

heroku create herodotbackend

Creating herodotbackend... done, stack is cedar-14

Git remote heroku added

Push your code to the heroku servers:

git push heroku master

Initializing repository, done.

updating 'refs/heads/master'

If you make changes on your code, al you need to do is to type "git push heroku master:master" again and your new code will be deployed.

**The AWS** is far from self -explanatory and documentation offered does not follow full scenario, it is just bits and pieces sewn together.

<https://eu-west-1.console.aws.amazon.com/vpc/home>

<https://eu-west-1.console.aws.amazon.com/ec2>

Let us assume that the instance is up and running, there is also another ceremony for you to get connected to the Debian server using a terminal.

You must generate a ppk from your pem file given to you by AWS using PuttyGen.

Load your ppk key to Putty and save it as a profile, with the user format something like;

[www.heredot.world](http://www.heredot.world)

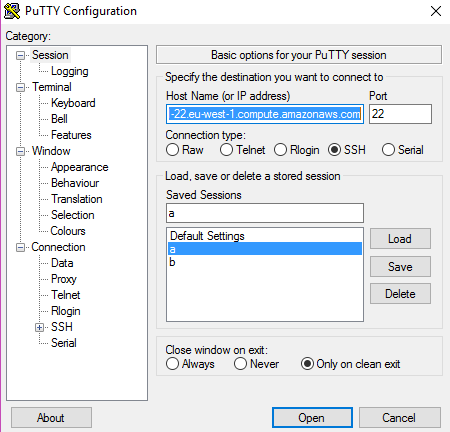


Figure 14 PuTTY Configuration Page

Once you are connected to the remote Debian server you must get the environment ready to install the Angular app through terminal cloning the github master branch of AyranIsTheNewRaki. Then the AWS will be run using the CLI and Angular files.

## Deployment Configurations of Front-end Application to Nginx on Debian

The issue of orphaned annotations after embedding Hypothesis client to our Front-end application. We assumed this was due to the domain name localhost. Instead of simply editing the hosts file to redirect requests from example.com to localhost:3000 I used nginx server.

1. Build the app

npm run build

1. Locate your default server path. Copy source directory contents and node\_modules into the server path

sudo su

cd /var/www/html

cp -R $Herodot/Frontend/src .

cp -R $Herodot/Frontend/node\_modules .

1. Edit nginx configurations to let the server route table know that if it can't find a file, path, it should direct the request to index.html. Contents of the file will be like the following:

server {

listen 80 default\_server;

listen [::]:80 default\_server;

root /var/www/html;

index index.html;

server\_name \_;

location / {

try\_files $uri $uri/ /index.html =404;

}

}

1. Edit /etc/hosts to redirect example.com to localhost:

127.0.0.1 localhost

127.0.0.1 example.com

# The following lines are desirable for IPv6 capable hosts

::1 localhost ip6-localhost ip6-loopback

ff02::1 ip6-allnodes

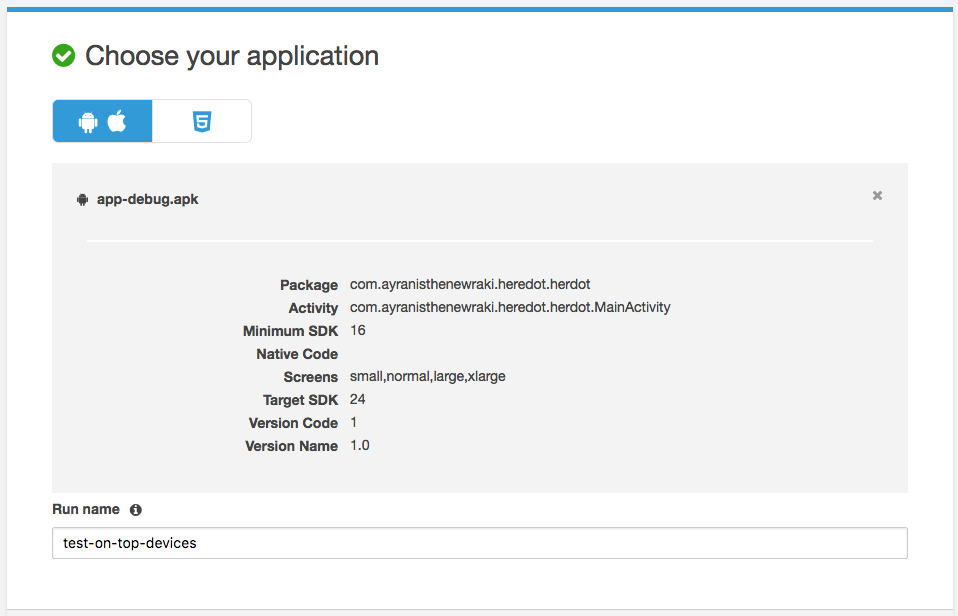
ff02::2 ip6-allrouters

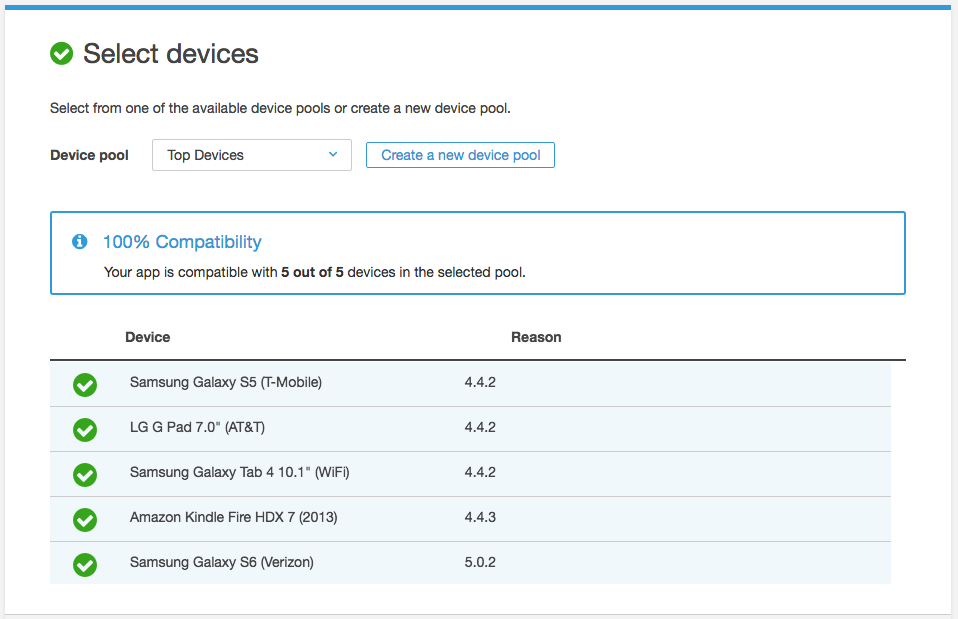
# Test Documents

**AWS Device Farm - Android Test Report**

Android devices come with great diversity, especially in screen size, but also in configuration. Therefore, the AWS Device Farm product was used to make sure the app builds and also looks good on top Android devices (top devices are the top most commonly used currently)

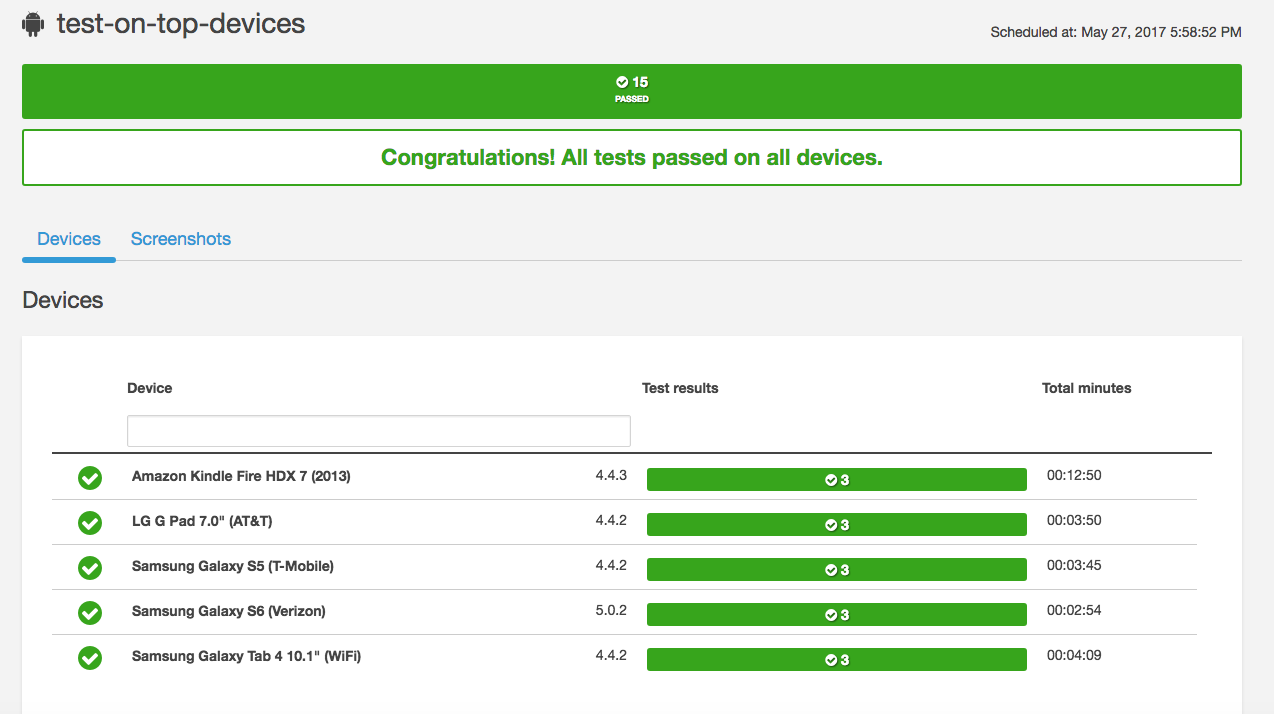
**Test Setup**: Upload APK and select devices to run test on.



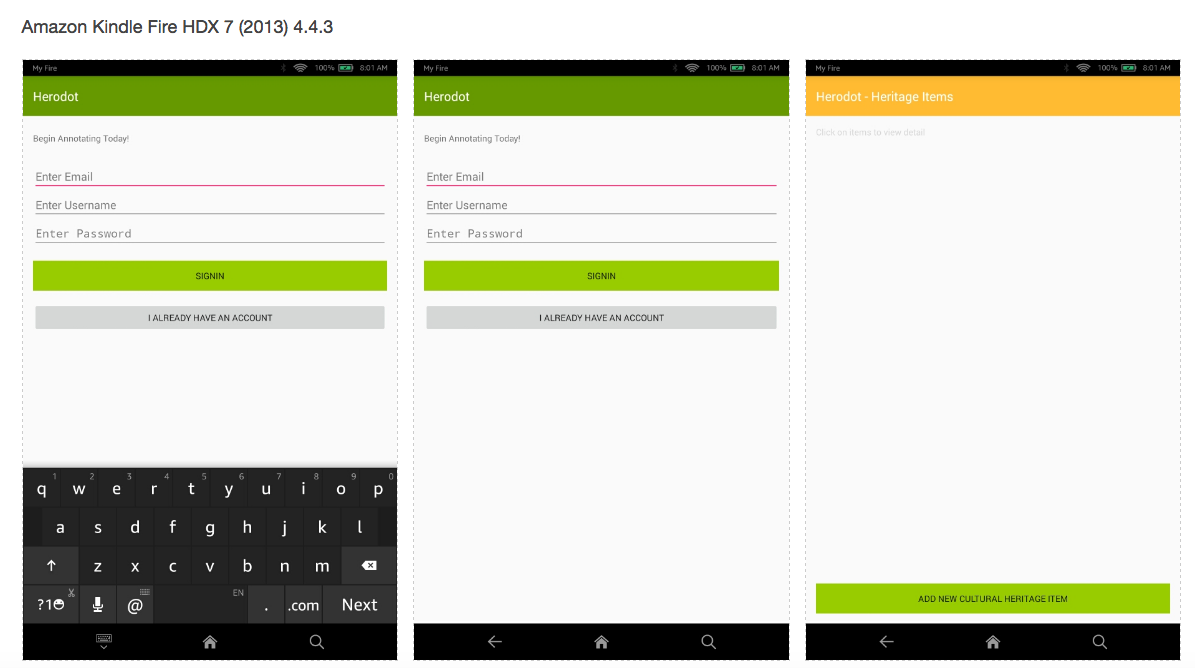


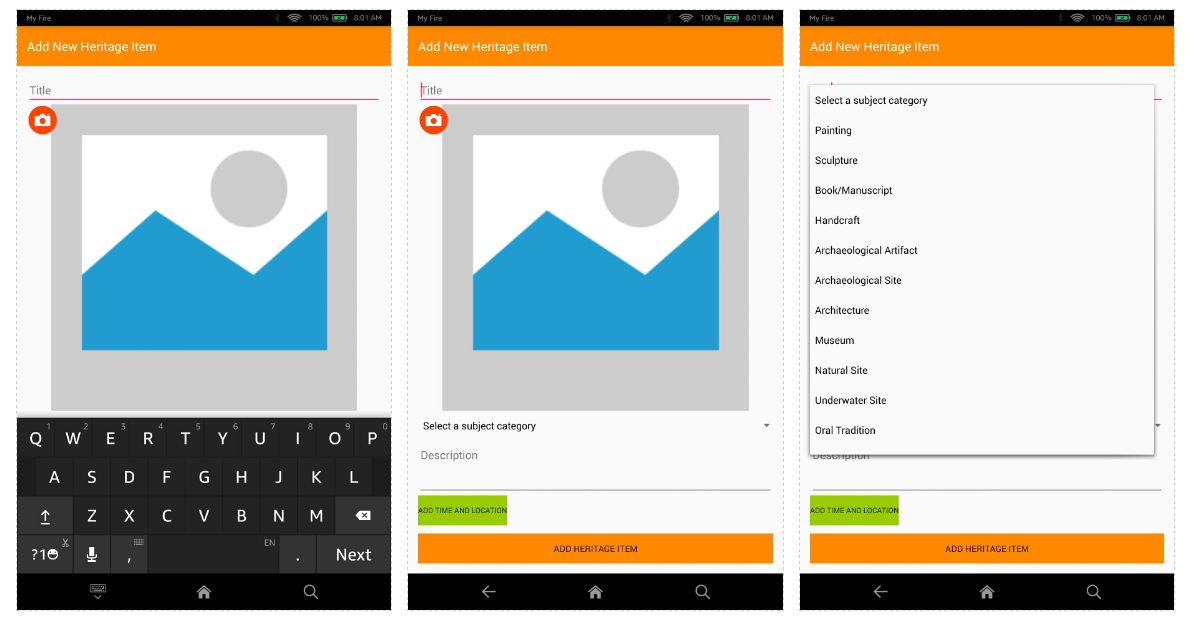
**Test Results**:

Build and Run Tests



Screenshots for screen size compatibility





# Tools and Technologies

The tools that are used during project life cycle (First experienced during SWE 573)

Below text is a place holder and a check list:

* Zen Hub: time management tool
* MS Project 2016: Project management tool
* GitHUb tags, issues utilized as tasks, milestones in line with the project plan
* DrawIO: mockups and SDD diagrams
* WebStorm: IDE for HTML, Javascript,CSS, ajax etc front end development
* Postman: RESTful API browser
* GitHub Desktop: to push and pull local dev files to github
* GIT: command line interface for version control
* PuttyGen SSH key generator utility: AWS pem to pkk conversion on windows
* Putty: connecting to the AWS web server
* WinSCP: synch files between AWS and local
* AWS web interface: deployment environment: EC2, VPC, IAM, CODEDEPLOY
* AWS Command Line Interface

# References

*Annotation Data Model*. (n.d.). Retrieved from https://www.w3.org/TR/2017/REC-annotation-model-20170223/

*DCMI Metadata Terms by Dublin Core Metadata Inititative*. (n.d.). Retrieved from http://dublincore.org/documents/2012/06/14/dcmi-terms/?v=terms

*Europeana as a annotation use case*. (n.d.). Retrieved from https://www.w3.org/2001/sw/sweo/public/UseCases/Europeana/

*Europeana Data Model Guidelines*. (n.d.). Retrieved from http://pro.europeana.eu/share-your-data/data-guidelines/edm-documentation

*Europeana Data Model Object Templates.* (n.d.). Retrieved from https://github.com/europeana/corelib/wiki/EDMObjectTemplatesProviders

*Europeana EDM Mapping Guidelines.* (n.d.). Retrieved from http://pro.europeana.eu/files/Europeana\_Professional/Share\_your\_data/Technical\_requirements/EDM\_Documentation/EDM\_Mapping\_Guidelines\_v2.3\_112016.pdf

*Europeana Linked Open Data*. (n.d.). Retrieved from http://labs.europeana.eu/api/linked-open-data-introduction

*Europeana Solid Example* . (n.d.). Retrieved from http://www.europeana.eu/portal/en/record/09102/\_UEDIN\_214.html

*Hypothes.is API creating annotations*. (n.d.). Retrieved from https://h.readthedocs.io/en/latest/api/#operation/createAnnotation

*Hypothes.is Blog*. (n.d.). Retrieved from https://web.hypothes.is/blog/annotation-is-now-a-web-standard/

*Hypothes.is document on security*. (n.d.). Retrieved from http://h.readthedocs.io/projects/client/en/latest/developers/security/>

*Pergamon Altar 3D*. (n.d.). Retrieved from http://3d.smb.museum/pergamonaltar/

*UNESCO.ORG*. (n.d.). Retrieved from http://www.unesco.org/new/en/culture/themes/illicit-trafficking-of-cultural-property/unesco-database-of-national-cultural-heritage-laws/frequently-asked-questions/definition-of-the-cultural-heritage/>

*W3.org Annotation Protocol*. (n.d.). Retrieved from https://www.w3.org/TR/2017/REC-annotation-protocol-20170223/

*w3.org annotations working group*. (n.d.). Retrieved from https://www.w3.org/annotation/

*W3C annotation*. (n.d.). Retrieved from https://www.w3.org/annotation/diagrams/annotation-architecture.svg

*W3C.org annotations announcement*. (n.d.). Retrieved from https://www.w3.org/blog/news/archives/6156