

Engage 0.3 beta Development Phases

Timeline

Dates	Design/Development Phase	McCord Preparation
Jan. 15 - Feb. 5	Open Coding	Prepare XML data, start podcasts, translate into French, setup WiFi
Feb. 8-12	Bug Parade	Finish translation and podcasts, setup iPod Touch devices
Feb. 15-19	QA Testing	Test deploy of Engage on McCord servers, help with QA, final deployment

Wireframes

[Mobile wireframes \(Draft 9\)](#)

Week by Week Plan

Week 1: January 18-22 (12 days)

Task	Estimate
Improve Data Imports	
Deal with HTML tags included in XML (e.g. from McCord)	3
Finish Exhibition Main, About, and Catalogue	
Code review and refinements to Exhibition components	2
Update Exhibit View to match Main and About wireframes	2
Update Catalog to match wireframes	3
Improve data access framework	
Refactor data access layer, removing redundancy across services (timeboxed)	2

Week 2: January 25-29 (12.25 days)

Task	Estimate
Add a Navigation Bar	
Implement Navigation Bar functionality for a variety of components	1
Add Bilingual Support	
Implement the language preferences screen using cookie support	0.5
Localization: ensure all components load strings in the correct language	4
Translate our strings to French (McCord)	0.75
Ensure all data feeds accept a language parameter and that results are returned correctly	0.5
Create Couch views to retrieve documents in one language or the other	0.5
Import a good, fresh snapshot of all the McCord data for the pilot and verify with Hugues	0.75
Improve Navigation	
Refactor as a "Thumbnail + Title + Description List" component (if necessary)	
Generalize Browse and Navigation List structure to use it in Exhibitions, Catalogue, and Browse by theme	
Add Object Code Entry Screen	
Implement Object Code component	2
Create Couch view for searching by code	0.25

Week 3: February 1-5 (13.5 days)

Task	Estimate
Improve Artifact View	
Update Artifact View to latest wireframes	2.5
Remove Tags from Artifact View (ditch it completely from svn)	0.25
Code review, refine, commit Sveto's changes to artifact that allows users to collect an item.	1
Finish My Collection	
Review Sveto's My Collection and plan next steps	2
Implement my collection email	1
Remove reordering from mobile version of My Collection	0.25
Simple login task that allows users to specify their email address for sending MyCollections	1
Make NavList restylable as a grid	1
Mock up Comments and Guestbook	
Implement comments list view	0.5
Implement comments entry screen	0.5
Integrate comments with Exhibit Main and Artifact View	0.5
Tying it all together	
Implement a home page with links to Exhibitions, My Collection, Object Code, Language Preferences, and Login	1
Work with Hugues to prepare for deploying and setting up the pilot	2

Assumptions

- Hugues will provide the following:
 - A set of XML files directly, containing all Exhibit and Artifact data we need
 - A service for emailing personal collection URLs to users
 - Integration between mobile My Collection and his CMS, so users on desktop browsers can see their collection
- We'll stick with CouchDB 0.9 instead of the latest version

Features We Won't Implement For 0.3

Object Code

- Implement Object Code component 2 days
- Create Couch view for searching by code 0.25 day

"See 25 more items"

- Share code with Paging to fetch data 0.5 day
- Write code to accumulate up a "growing model" and re-render the component 1 day
- Add support for refreshing views in NavigationList 0.5 day

Next/Previous Artifact in Artifact View

- Fix paging to open at a particular page 0.5 day
- Update ArtifactView to use this, and integrate into NavBar 1 day

Desktop My Collection

- Implement desktop web version of My Collection 3 days

Performance Enhancements

- statically make a custom build and commit alongside Engage 1.5 days

User Experience Polish

- Fix Safari touch delay if necessary 3 days

- Improve error messages 0.5 day
- Generalize spinner and loading page from Browse 1 day

Guest Book and Comments

- Implement comment saving 2 days

Login and Sign Up

- Implement the wireframes as components/html pages + JS 1 day
- Need to add Kettle API for generating password hashes 1 day
- Add support for HTTPS 5 days
- Ensure user information is stored in a cookie 0.5 day
- Design a strategy for identifying users: 1 day
 1. Museum device: for the duration of using the device (cleared upon quitting the browser)
 2. Personal: forever (unless user clears their cookies)

QA Test Plans

- Create test plans for Voice Over screen reader with iPhone 2 days
- Create test plans for all new functionality and update for existing screens 2 days