

Embodying Compassion: Project Plan

Members: Clive Mudanda, Gary Sheng, Anthony Olawo

Executive Summary

Mission of application

This app is meant to enhance the first transcultural exhibition in America solely devoted to Bodhisattva Avalokiteshvara. It will serve as an onsite guide for those who have the privilege to experience the Bodhisattva exhibit by providing contextual content. And, this contextual content will also provide those unable to attend the exhibition in person a remote gateway to experience it, increasing the number of people impacted by the exhibit.

Functionality overview

The app launches at its title view. It'll then transition to a loading view prompting the user to enter. Upon entry, the user will decide to either enter the explore view or search for a particular artifact's view.

Explore View

From this view, the user will be able to access any of three theme views that mirror the actual exhibit's themes.

Theme View

This view will contain a grid of titled thumbnails, each linking to an artifact view. Tapping on a particular artifact will bring up its artifact view.

Artifact View

Each artifact view will be uniquely identifiable by a number and QR code linking it to the actual exhibit artifact. Each artifact view will also contain a bucket of rich multimedia context in the form of concise captions, narrated explanations, and visuals (either high resolution stills or video) showing you a particular artifact in detail. It will have:

- an **audio guide** enunciating an artifact's title.
- a concise and contextually informative **caption**.
- a **multimedia section** containing videos and audio clips (e.g. short films related to pilgrimage sites and rituals, recordings of music, and chanting) related to the artifact.
- a zoomable **full screen image view** that'll fill the screen when you tap on the image.
- where it applies, an artifact view should have a gallery of comparative images. See **Comparative Images View**.
- a **map** locating the artifact's geographical origin.

The app will also have sidebar navigation that's accessible from every view on the app.

Comparative Images View

This will be a full screen view. It'll show images from different cultural contexts to be compared to the original artifact. Each image will have an overlaid caption. The user will be able to see the next or previous image in this comparative images gallery by swiping left or right.

Sidebar Navigation

As the primary navigation tool, it'll allow the user to jump to the about view, the explore view, slide show view, and terms view. It'll also allow the user to search for an artifact by number or name.

About View

This view will contain text explaining the purpose of the exhibition and app.

Slide Show View

The slide show view offers the user a chance to experience the exhibit's artifacts outside of its thematic organisation by presenting a randomized slide show of artifact views from each of the three themes.

Search Results View

This view will be accessible by tapping the 'search by artifact' button on the loading screen or the side navigation bar. It will give the user an option of either typing out the title of the artifact or scanning the physical artifact's QR code to find it. Upon successfully executing the search, this view will show titled thumbnails of artifact views that match the search query entered.

Technical Considerations

We settled on a cross platform mobile app to be built on [ionic](#), "a beautiful, open source front-end framework for developing hybrid mobile apps with HTML5," CSS, and Javascript (all web development languages). We will be couching the app in [phonegap](#), another framework that'll will compile the app in the cloud and churn out downloadable versions of it for iOS, Android™, Windows® Phone, Blackberry® 5/6/7 and webOS to ensure that we reach the largest possible audience.

That said, we will be storing the bulky multimedia on a remote on a [Parse](#) backend to keep the downloadable app light; we'd like it to download onto the user's phone as quickly as possible.

Overall Goals Scope

Goals

Our primary goal is to deliver an app that will provide contextual information for those visiting the actual exhibit while acting as a virtual exhibit for those unable to attend. To this end, we will incorporate our client's feedback on each iteration of the app into the next iteration while prioritising critiques of the app's functionality, before critiques of its user interface design. However, once we achieve this goal we will reach for our stretch goal.

The stretch goal will be to make it possible for the client to manage the content on each instance of the app from a remote location. This would mean developing a web app through which an 'exhibit owner' can add, remove or edit content and have those changes reflected on the mobile app instances linked to her/his exhibit.

Design

With the stretch goal in mind, we are designing to ensure that the mobile app's content can be edited by individuals without programming experience. The title of the exhibition and its about page should be editable remotely. Secondly, the manner in which content in the app is organised as well as the actual content itself ought to be editable as well; at the moment, this means that one should be able to shift artifact views between various themes, and edit the content of the artifact views themselves. The content of the glossary should be editable as well.

As such, the core features for the mobile app will be the thematic organisation of artifact views, the ability to search for a particular artifact's view, a glossary containing key terms and definitions, and the ability to view a slideshow of all the artifacts in random order-- disregarding their thematic organisation. Lastly, and most importantly, there will be a side navigation bar for easy intuitive access to each of the views.

We will implement the following features of the artifact view as soon as the core features are fully functional: an **audio guide** enunciating an artifact's title; a concise and contextually informative **caption**; a **multimedia section** containing videos and audio clips (e.g. short films related to pilgrimage sites and rituals, recordings of music, and chanting) related to the artifact; a zoomable **full screen image view**, that'll pop up when you tap on the image; a gallery of comparative image and a map locating the artifact's geographic origin.

This project's main dependency is the content the client will deliver: text, high resolution images, video and audio. However, the team does have concerns. We will be setting up a schedule for the client to submit content to us in our fourth meeting of the semester.

None of the team members have used either [phonegap](#) or [ionic](#) before. Therefore, we are anticipating a slight lag at the start of the project as we ascend the two platform's learning curves. In addition to this, we are also concerned about app speed tradeoffs: should we store 'heavy' multimedia content remotely and reduce download time while incurring latency costs each time a user wants to, for example, watch a video? Or, show we store them on the app, increase download times, and avoid incurring the above latency cost? In that regard, we'll also need to find out from the client what the wifi speed at the exhibition is.

Team Organisation

Being a three man team has pushed individual team members to take on dual roles. **Clive Mudanda** will be our Team Lead. As Team Lead he will be maintaining the project's milestones and timeline, ensuring that the team has all the necessary resources, while also communicating the project deliverables to both the client and class instructors. As Business Associate, he will also be the client's primary contact, coordinating client meetings, managing the requirements put forward by the client during those meetings as well as representing the client's needs and point of view to the rest of the team when the client is not present. **Gary Sheng** will be our UX Design lead. The client should contact him with specific questions regarding the look, feel and ease of use of the app. Lastly, **Anthony Olawo** will be tech and Quality Assurance lead. As technical lead, his role will be to break down the project into features that the rest of the team can code as well as identify what tools would be best suited to coding these features. Here he will liaise with the client on the impact that using certain tools would have on the app's overall performance, amongst other things.

Deliverables

Based on the goals and priorities outlined above, this is what we will deliver for each of the sprints:

Sprint 1.0 pretotype

For our first sprint we showed our client a pretotype built using Xcode's storyboarding feature. The client gave us extensive feedback that we incorporated into Sprint 1.1

Sprint 1.1 pretotype

We went back to the drawing board for this Sprint. The team designed a set of mockups that we easily pretotyped [here](#). We showed both the mockups and the pretotype to the client who remarked at the quick turnaround and offered more feedback that we used to tweak our pretotype for Sprint 1.2.

Sprint 1.2 pretotype

This Sprint linked sharper mockups via smooth transitions to create a cleaner user experience. [Here](#) it is, a pretotype.

Sprint 2 prototype

We'll code out a skeleton of the that closely matches Sprint 1.2 for this Sprint. It'll contain all the views that the final app will have as well as dummy content. But, its functionality will be limited to navigation.

A list of the views:

1. the title view that leads to
2. the loading view that'll hold until all resources are loaded before transitioning to
3. the 'enter the exhibition' view that leads to

4. the explore view or the search results view, depending on which of the two the user chooses.
5. The explore view will contain three theme views each of which will house a grid of
6. artifact views each linking to a comparative image view.
7. Most importantly, there'll be side navigation to allow movement to the about view, slideshow view as well as the explore view and search results view.

Sprint 3 baseline

This sprint will extend the previous deliverable to include search for artifact views and search for key term views functionality. It'll also include an algorithm to randomise selection of artifact views for the slideshow view.

We'll also deliver on the following features of each artifact view:

1. an **audio guide** enunciating an artifact's title.
2. a concise and contextually informative **caption**.
3. a **multimedia section** containing videos and audio clips (e.g. short films related to pilgrimage sites and rituals, recordings of music, and chanting) related to the artifact.
4. a zoomable **full screen image view** that'll pop up when you tap on the image.
5. Where it applies, an artifact view should have a gallery of comparative images. See **Comparative Images View**.
6. a **map** locating the artifact's geographical origin.

This app will contain actual data provided by the client.

Sprint 4 alpha

To finish off the app off we'll implement QR code search.

Depending on time constraints, this Sprint will include our stretch goal deliverable, which is a web app through which the client will be able to edit the content of the app, and move it around different themes.

While this Sprint will be buggy, we'd recommend the client deploy it in the exhibition and give the team feedback on the experience.

Sprint 5 beta

This version of the app will be potentially bug free and ready for deployment by the client or stress testing by multiple users. It'll be bundled with a document containing a description of the application, and overview of the test plan, instructions on how to conduct the stress testing, and a feedback survey to conduct after the stress testing,

Sprint 6 release

This iteration will be a robust extension of Sprint 5 to account for any bugs caught during user testing. It'll be bundled with technology transfer documentation needed by the user to use,

maintain and upgrade the app after the development team is disbanded. The documentation will contain a user manual, an installation and maintenance document detailing how to manage the app and a comprehensive project code design document for future programmers consisting of.

The project code design document will consist of:

- a. architecture, design, DB schema, technology inventory, toolkits
- b. instructions on how to create dev environment, access code, build & deploy
- c. in-code comments (including your names!)

The documentation will also include:

- d. an about page in the app with the team member's names, clients' names, Duke University, and any other supporters your client has had in their research
- e. a README.txt file in your project's top level folder that explains your project's purpose, its dependencies, and how to deploy your project
- f. a RELEASE_NOTES.txt file that lists your project's current functionality and what platforms it has been tested on
- g. a BACKLOG.txt file that lists future features you would like to implement
- h. a LICENSE.txt file that describes your license choice for the project (mostly this is boilerplate text from [the license's official page](#))