# Android Art @ GV

## Design Document

Aaron Merritt, Jon Bardsley, Mark Minuth, Sari Rahal

9 February 2012

# GVSU Design Document

## Introduction

The content of this document is to illustrate design ideas that are being used in the Art at GVSU Android App. The application has a variety of features including tours, browsing, searching, and favorites.

The tours section utilizes google maps APIs to indicate where artwork is within a tour. This feature uses your location as a reference and enables users to get details about specific artwork. Due to some current server issues, the tours list is unable to load icons, but can still be called using the web service.


Picture 1: Tour Example 1

Picture 1 is an example of the Outside Sculptures tour on the GVSU Allendale campus. The image displays the key points of the tours and where they are located using the Google maps features for mobile applications.

Both the browsing and search features allow users to find artwork by building location, artist name, image name, or image id number. Once users reach the artwork's detail page (Picture 2) they can choose to add the artwork to their favorites list.


Picture 2: Favorite Example 1

As you can see in the photo on the right, the heart is highlighted; by using this touch feature a user can add the current art piece to their favorites list. This lets them gain fast and easy access to it later.

Each piece of artwork is stored in the open source collection management system, Collective Access. In order to get information from this relational database the service is called using this request: http://gvsuartgallery.org/service.php/iteminfo/ItemInfo/rest with the following parameters:

Method = Method name

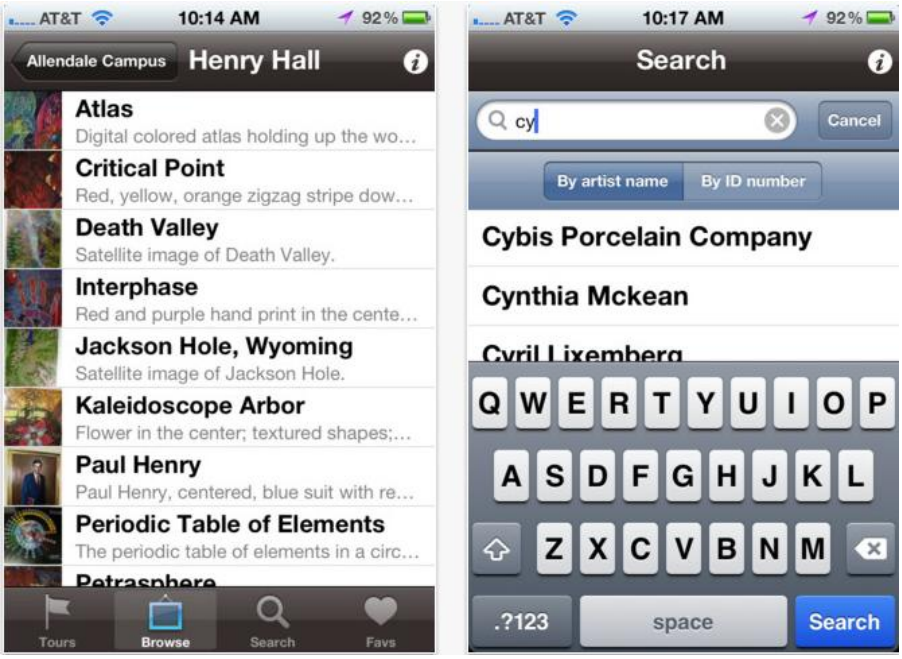
Type = object type in Collective Access framework

Item_ids[]= Array of numeric Ids, 1 will bring the info of all campuses

Bundles[] = Array of possible fields

For example the web call bellow will return a list of campuses and other facilities. After parsing the XML data if the value is not a campus then it will be filtered out.

http://gvsuartgallery.org/service.php/iteminfo/ItemInfo/rest?method=get&type=ca_storage_locations&item_ids[0]=1&bundles[0]=ca_storage_locations.children.location_id&options[ca_storage_locations.children.location_id][returnAsArray]=1&bundles[1]=ca_storage_locations.children.preferred_labels.name&options[ca_storage_locations.children.preferred_labels.name][returnAsArray]=1




Picture 3: Browse and Search

While the user is putting in their request the application is suggesting art names in an unordered list below. Shown on the left in Picture 3 is the browse feature where you can chose which campus, area, etc to look through.

## Language and Frameworks

Android 2.3 OS or Gingerbread is the most widely used for the Android based phones. Because of this we will be writing our application using the 2.3 API. We believe most users are donned with 2.3 on their devices because of the large leap it made from previous versions. The enhanced screen support, interfacing, power management, enhanced keyboard, Google maps and many other functions were key in the decision making as well.

We will be using the Eclipse IDE (Integrated Development Environment) along with NetBeans. Android is coded in Java and XML. In order to develop Android apps with Eclipse, the ADT (Android Development Tools) is also required. This plug in extends the capabilities of Eclipse to give easy implementable features for creating applications, UI, and components based on the chosen API, in our case 2.3.

In order to give better interfacing to our users we will be accessing the Google libraries while creating many of the mapping features. These will include things like the touring and location features.

## Organization

The GitHub repository will have separate Wikis; journals for each team member will be kept here. Also, links to each document, as it is worked on and completed, will be posted in the wikis page.

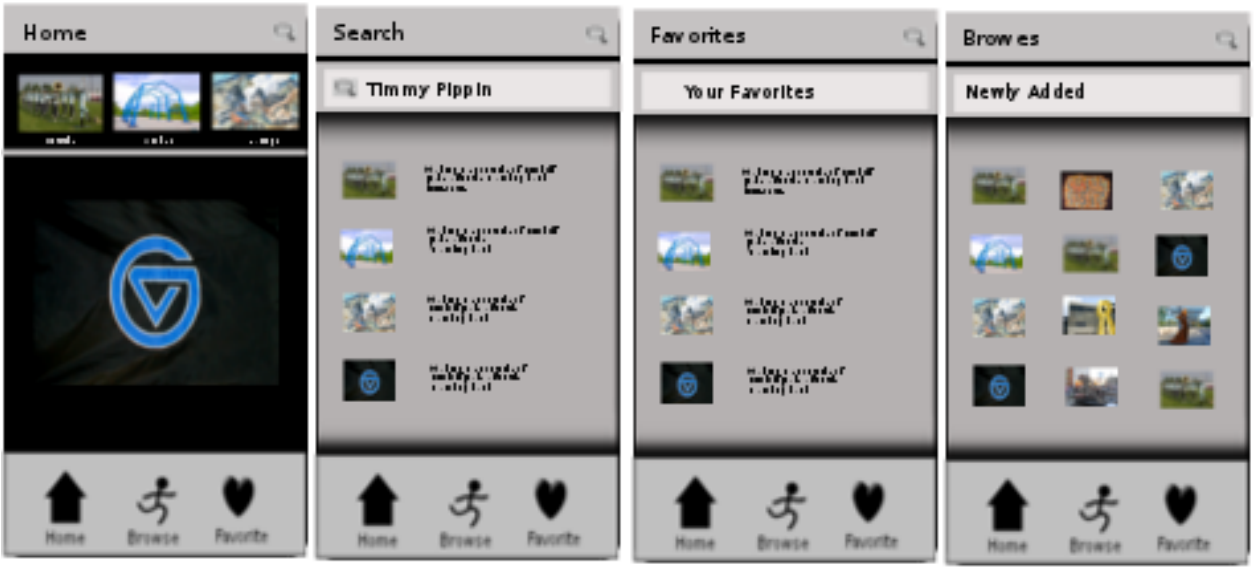
Each member is responsible for a section of the application, described below. Each week there will be a team meeting where the group will gather the current information and go over the details of their progress. If possible the different pieces will also be integrated together, making sure the pieces all still fit before moving on to the next phase.

Also the team will be trying to set up a bi-weekly meeting with the GVSU Art team in order to keep them informed and gather input on the most up to date phase completions.

## Labor Division

Since there are four distinct parts to the Art at GVSU app, labor will divided equally. These parts include: Tours, Browsing, Favorites, and Search. In addition to these features, the information page (Picture 2) will be implemented and utilized in each section. Sari will work on this along with the favorites page. Jonathan will work on the tours and home page, Aaron will code the search feature, and Mark will write the browse portion of the app.

Design of the user interface has been a collaborative effort, as Nathan has requested that all features remain similar to the current IOS version. Below are images of possible interfaces for each of the four main features. The mock ups show a menu bar at the bottom for users to return home, browse artwork, and access their favorites. Our decisions for the design are based on the fact that most android applications use large menus, which we want to stay away from. It is important to us that users experience an interface that is Android specific but does not limit the functions of the app. Similar designs to the IOS app will be used while making sure to adhere to Android users and devices.




Picture 4: Android Designs

Above are the rough depictions of what we are heading towards. These in no way represent what we plan the final iterations to look like; more of a first draft depicting some of the features we are currently thinking over. The main points to take away are the two bars at the top and bottom of the screen. The bottom bar is there as navigation while the top will be a search and information tool bar.

Although these appear similar to the iOS applications we will be trying to integrate them to have a more modern Android version feel. After consulting with Dr. Andrew Kalafut and Dr. Hans Dulimarta we have settled on much smaller, less abrasive tool bars moving towards what Google is trying to push. We have looked at similar apps for Android, such as “Buildings”, and decided that similar features will also be used.

## Database and Memory

The database that will be used is Collective Access. This enables web service to return XML information when requested through search features or other aspects of the app. The XML data will then be parsed and displayed to the user in the specific location. Below is an example of the returned data from a search for the artist name “Tim”.


```
<CaSearchResult>
  <ca_entities entity_id="1260">
    <displayLabel>Tim Nowakowski</displayLabel>
  </ca_entities>
  <ca_entities entity_id="1289">
    <displayLabel>Roger Timermanis</displayLabel>
  </ca_entities>
  <ca_entities entity_id="1339">
    <displayLabel>Tim Kennedy</displayLabel>
  </ca_entities>
  <ca_entities entity_id="1368">
    <displayLabel>Tim Fisher</displayLabel>
  </ca_entities>
  <ca_entities entity_id="1666">
    <displayLabel>Tim Lowly</displayLabel>
  </ca_entities>
  <ca_entities entity_id="1958">
    <displayLabel>John Timothy Pizzuto</displayLabel>
  </ca_entities>
  <ca_entities entity_id="2031">
    <displayLabel>Timothy Norris</displayLabel>
  </ca_entities>
</CaSearchResult>
```

The database for the application will be implemented using a built in feature called SQLite that is available on all Androids. SQLite is an Open Source Database that accepts standard relations such as SQL syntax while keeping the demand for memory low during runtime. After we define the SQL statement for creating/updating a database, the Android platform automatically manages the database.

The application will also allow the users to create a favorites list that will be organized in the database. Access to a SQLite database involves accessing a file system. This can be vary memory demanding and slow. We are currently looking into ways to reduce the demand of memory thus speeding up access.

One of the issues that we are looking into is the catching of loading images. Andres had warned that he had some issues with the program reconnecting and reloading images that were already loaded on the device. This caused the program to use a lot of memory and drain the battery rather fast. Andres mentioned that by storing some of the information on local storage will help both the speed and battery.

## Testing and Gathering

In order to create the best application, we have and will continue to browse through the Android market place for insight into what is 'current' and gathering the most 'buzz', as we are trying to make the application as appealing as possible.

Once we have gathered our UI design ideas there will be a time for re-design and 'light' testing. If the group finds any caches they will be altered here.

During the final weeks of development the group will be putting aside time for final testing and black box testing. The objectives will be to ensure that all of the ideas and features are working correctly before public use. Seeing as there are different OS versions for mobile phones and tablets, most of the testing will be done for the phone portions; saving tablet testing only if there is time available. As the group currently does not have access to a large group of testers most of the testing will be done 'in-house' or within the group.

Currently the group is unaware of any protocols that would hold any of the features being implemented.

## Timeline and Development Breakdown

Team meetings will be held weekly and Nathan will be included in bi-weekly meetings. We will consult with Andres when technical questions need to be answered and discuss changes to the user interface with Nathan and all other parties.

Appendix A holds Gantt charts from each team member, depicting the beginning and ending phases for their sections. As each team members phases come to an end the finished member will move on to help the other members as needed to help keep the project on target.

Appendix B holds the architecture breakdown for how the group views the project at a high level.

# Appendix A

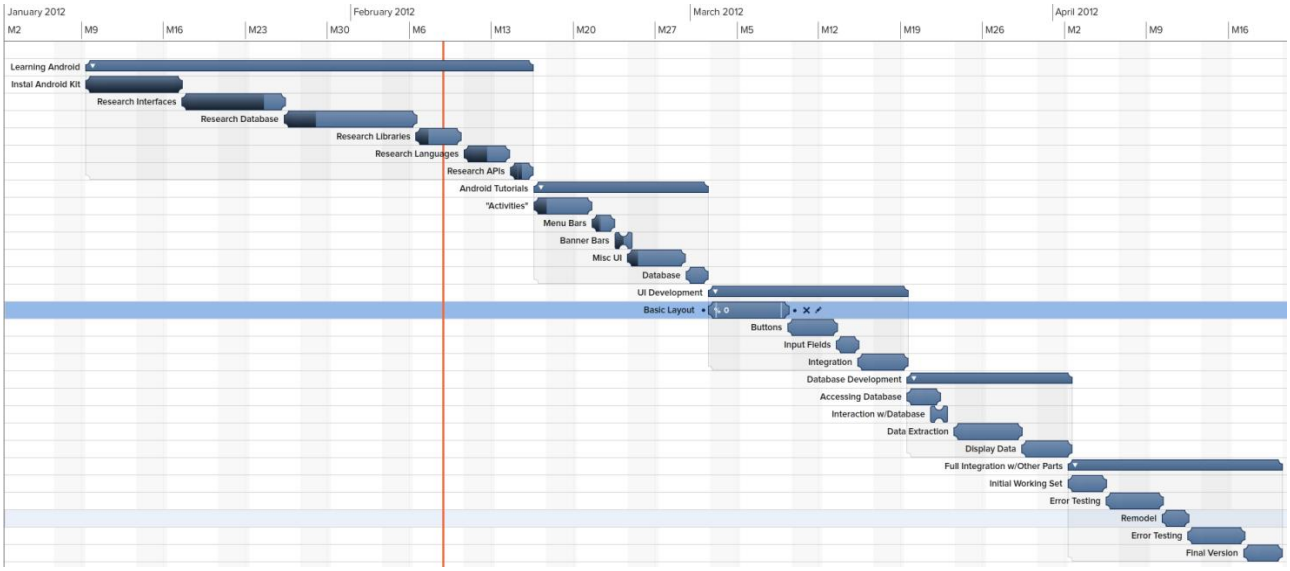



Figure 1: Aaron's Gantt

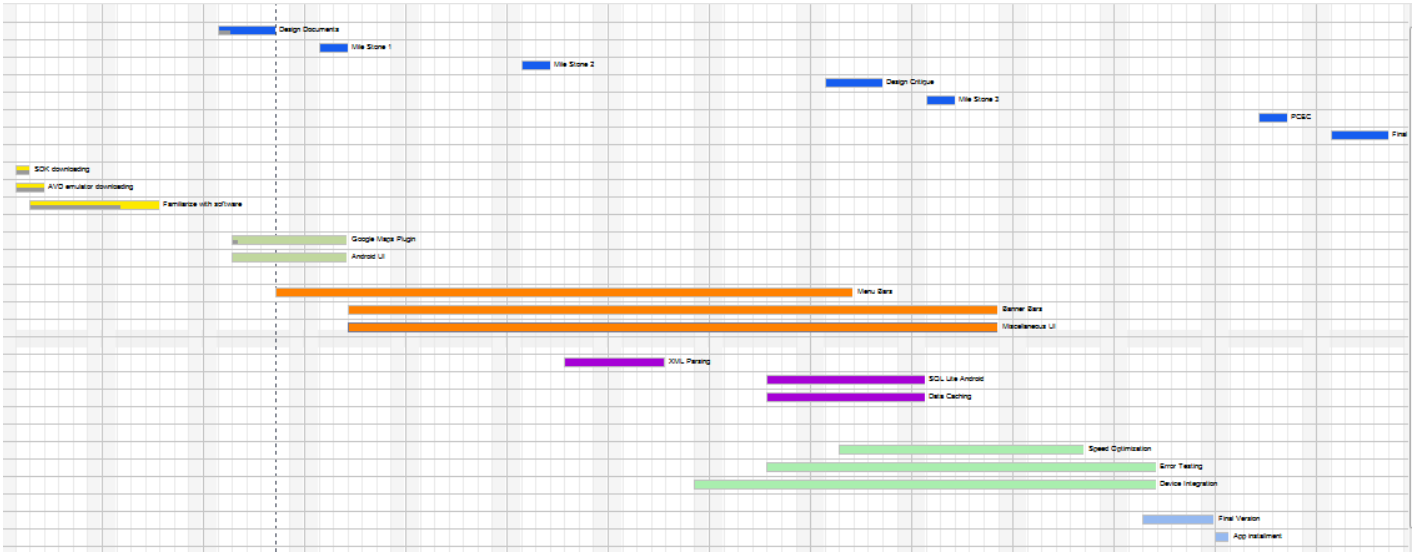



Figure 2: Jon's Gantt

# Appendix B

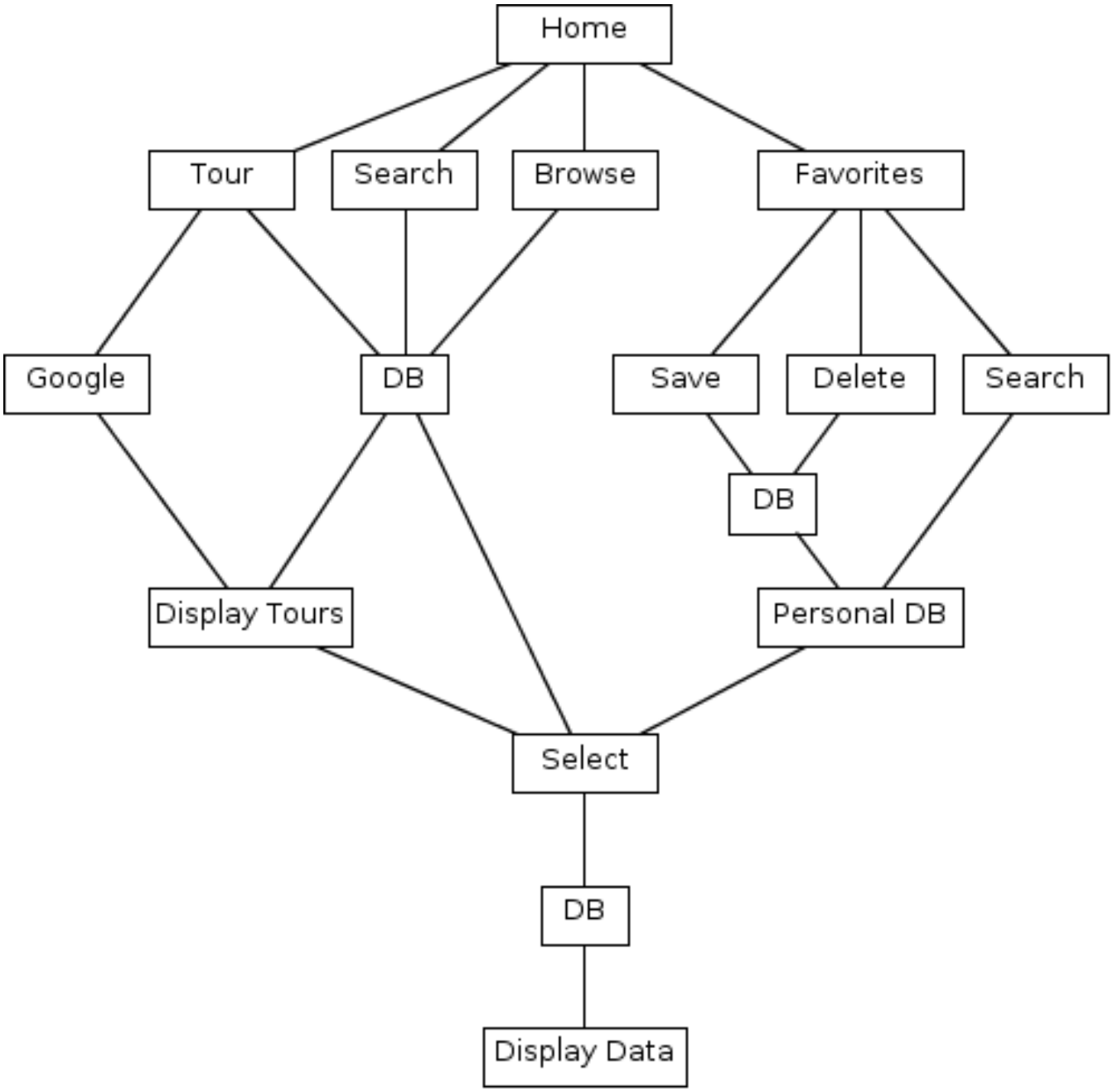



Figure 4: Architecture