MODS Explorer Application

Amarnath Maharaj, Kyle Martinez, Bradley Zarek

**Abstract**

The Museum of Discovery and Science is in need of a mobile application that will bring the museum to the forefront of innovation. A museum application can have numerous benefits to both the museum and the visitors of the museum. Primarily, a mobile app can supply information rapidly and efficiently. On an educational level, an app can enhance the experience of the user by providing descriptions of the exhibits. By using an interactive map, the app can improve the overall quality and efficiency of the visit on a navigational level. Additionally, the museum staff can gain insight into which exhibits are popular with the visitors. When creating our app, our goal was to cover each aspect and integrate them into the features of our app. By using a combination of Java and Processing, we successfully built an app that contains a wide array of features, accomplishing our goal. To further increase the interactive portions of the app, we added user accounts with user data uploaded to Parse and collectable stickers that appeal to younger viewers.

**Background**

After a visit to the museum, the exact needs of the museum were clear. A common issue experienced by visitors was the fact that many people did not get a full experience by simply reading the exhibit information. Some parts of the exhibits had no description at all, which does not help with learning and discovery. This common problem led us to create our app with an intention to be educational to those who use it. It also gave us the idea to include both images and text to convey the information so that people at the museum can easily identify and learn about the exhibits.

Another suggestion that we received from the museum tour was the idea for a virtual map. Despite the museum’s desire to innovate, the map of the museum is still printed on paper. This motivated us to include an interactive map as part of our app. As an added benefit, a virtual map eliminates the waste of paper and helps the environment. We also decided to include a “stickers” page in order to inspire younger users to explore the museum.

**Method**

The design of the MODS Explorer application contains a welcome screen, a home page, and several subpages. The welcome screen and home page are written in Java, the subpages are written in Processing, and assets are referenced for graphics. The welcome screen consists of an option to either create an account or log into an existing account. When a user chooses to create an account they will be prompted for information such as their username, password, email address, and age. If any of the information is found to contain errors, the user will be notified immediately and the registration process will be temporarily halted. Once the user’s information is found to be error-free, and the “Sign up” button is pressed, a new ParseUser object will be created in memory. This object contains all of the information provided by the user, along with a HashMap<String, Boolean> containing the user’s sticker collection. At this point in time all of the Boolean values stored within the map are false, as no stickers have been collected. With the ParseUser object created, the information is then uploaded to Parse.com to be stored in a database table. Once this process has completed the user is returned to the welcome screen.

To access their newly created account the user must now log in. Having selected the “Log in” option on the welcome screen, the user will be prompted to enter their username and password. As with the registration screen, any errors found in the information will be immediately reported to the user. Once logged in, the user is brought to the home page where there are buttons linked to various subpages. These subpages include an interactive map, an event listing, a sticker page, as well as an NFC detection page.

The map page loads images for the two floors of the map and the exhibit images as PImage types. The draw() method contains an if statement that switches between the first and second floors. If the integer variable mapToggle equals 1, it displays the first floor map. If mapToggle equals 2, it displays the second floor map. Once the map is displayed, the user can drag and pinch the screen to move and zoom into the map. The mouseDragged() and onPinch() methods allow for this function, changing the values of the mapX and mapY coordinates (position of the map) as well as the mapWidth, and mapHeight variables (dimensions of the map).

When the user taps the screen, the onTap() method runs. This method contains an if statement for every interactive portion of the map, including each exhibit and the floor number button. By using a combination of the position, dimension, screen width, and screen height variables, the if statements calculate the lower and upper bounds for both x and y. If the mouseX and mouseY values of the tap fall between those values, the tagValue variable is set to the number of that exhibit. In the draw() method, a switch statement switches among each possible value of the tagValue variable, displaying the correct exhibit page when the map is tapped.

Similar to the map page, the NFC detection page is intended to display each exhibit page. It contains the same switch statement and the same tagValue variable, but the value of that variable is set differently. An onNFCEvent() method runs whenever an NFC tag is detected, and inside the method is a line of code that sets the tagValue equal to the integer value of the tag. This page achieves the same result as the map page, but it is intended to be easier for people who are in the museum and want quick access to the exhibit page.

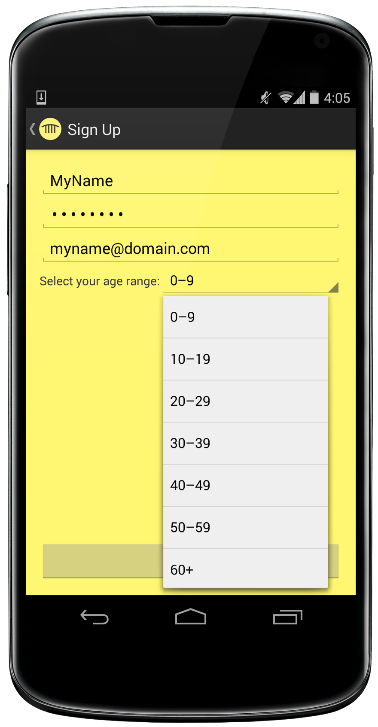
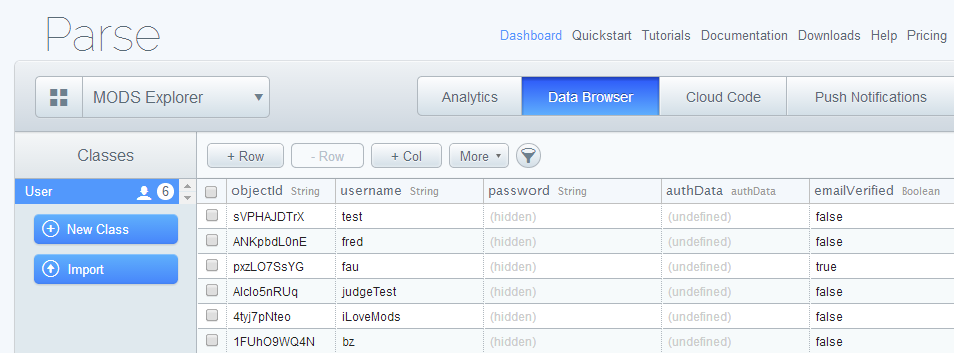
Both the map page and NFC page display separate pages for each exhibit. For each value of the tagValue variable there is a switch statement that allows the user to navigate through different parts of the exhibit. By clicking the arrows at the bottom of the screen, the imageNumber variable is either incremented or decremented depending on which arrow is clicked. The app displays a picture and text corresponding to the value of imageNumber. If the right arrow is clicked, and the imageNumber equals the pictureCount (the app is displaying the last picture of the exhibit), the imageNumber is set to one. If the left arrow is clicked and the imageNumber equals one (the first picture), the image Number is set to the pictureCount. This allows the user to view each image and loop back to the beginning once all pictures have been displayed.

The last page of the app is the stickers page, which simply displays the stickers that have been collected. Stickers that have been collected appear in color, and stickers that have not yet been collected appear in grayscale. The stickers are displayed on three pages that loop similar to the exhibit pages. By clicking the arrows, the user can view each page of stickers and continue to click to loop back to the beginning. After the user scans the NFC tag at the exhibit, the sticker for that exhibit is collected and uploaded to Parse. That data is retrieved when the stickers page is viewed.

Each page of the app utilizes assets. Most of the assets were handmade images using programs such as Adobe Illustrator and Photoshop. Of the assets that were not handmade, only images that were labeled for reuse with modification or with no copyright were used. The handmade assets were made in Adobe Photoshop or Illustrator by first using multiple shapes merged together to make an image. Then, using the blending options format, bevel, and any other effects deemed necessary were added.

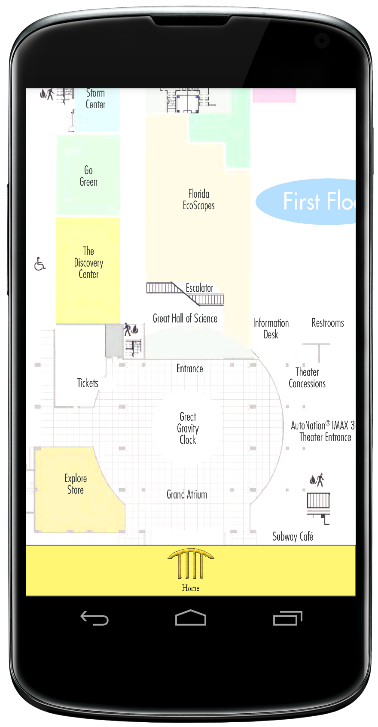
**Results**

We were successful in creating our application, and we achieved each goal that we set. When the app is launched, a welcome screen allows the user to either log in or create a new account. When the user chooses to create an account, a screen appears that asks for a username, password, email, and age range. Each of these fields is uploaded to Parse, which stores all account data.



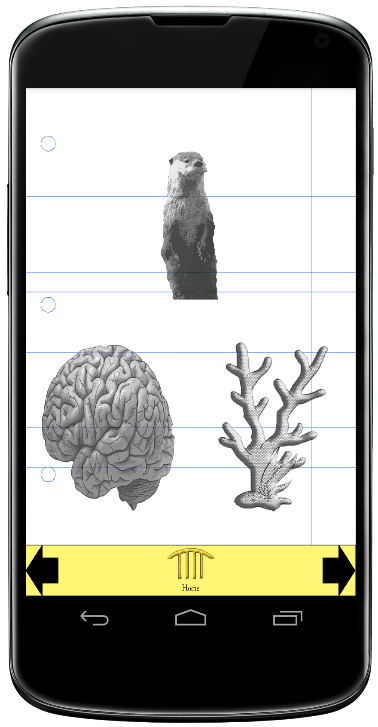
*The interface that the user sees along with the Parse data browser*

We also successfully created a welcome page subpages that are displayed after the account information has been entered. From the home page, the user can navigate to the map page and view the interactive map. The user can also click on an exhibit to view the exhibit page. Alternatively, the user can swipe an NFC tag to reach the same exhibit page.



*The interactive map page and an example of an exhibit page (Florida Ecoscapes exhibit)*

We were also successful in creating a stickers page that retrieves information from Parse. The user can navigate through each page of stickers, and only stickers that have been collected (by scanning the correct NFC tag) are displayed in color.



*The stickers page, which shows the Otter, Powerful You, and Florida Ecoscapes stickers (not collected)*

**Discussion**

During the process of creating this app, we faced many challenges. The first issue that we had was the integration of Processing code into Java code. At first, our idea was to run the Java parts of the app while still displaying graphics such as a menu bar from the Processing code. After several attempts, it was clear that this method did not work. However, we quickly decided that a simpler way to accomplish our goals was to run different pieces of the Processing code from a home page written in Java. This attempt was successful. Another issue that we faced was the memory of the mobile device. By loading many pictures in the app, a large amount of memory was used and created errors. We solved this by editing the sizes of the images, which decreased the amount of memory that was used.

**Conclusion**

Our app not only accomplishes each of our original goals, but also opens the opportunity to add new features to the app. The app currently contains the feature of user accounts and collectable stickers, but this function can easily be expanded. The map page, which already is interactive and educational, could potentially suggest museum exhibits and activities based on the user’s account information. The home page could have an option for the user to send information to the museum staff regarding that user’s visit. The app has a high potential to improve both the experience of the user and the connection between the visitors of the museum and the museum itself.

**References**

Exhibits. (n.d.). Retrieved from Museum of Discovery and Science: <http://www.mods.org/>

Images that were used for assets can be accessed from the following links.

<http://www.aquila-style.com/fashionbeauty/bizarre-beauty-routines/46739/>

<http://commons.wikimedia.org/wiki/File:Atlantis_taking_off_on_STS-27.jpg>

<http://news.anu.edu.au/2013/12/06/changing-environments-a-challenge-for-the-brain/>

<http://thewisenest.com/wp-content/uploads/2013/02/ocean-zones-WN.pdf4>

<http://www.dragoart.com/tuts/10760/1/1/how-to-draw-a-dinosaur-skeleton,-dinosaur-skeleton.htm>

<https://sites.google.com/site/6banimaiseuropa/animaisterrestres>

**Appendices**

The MODS Explorer Application can be found in a zip folder, and there are separate zip folders for the assets that were used and the separated Processing code.