Android Final Project

App Name: Yamsterdam

Timothy Yao - ly3py  
Jiu Moon - jkm4ca  
Android:

Project Pitch:

Due to the increasingly busy lifestyle adopted by modern society, leisure time has become more scarce, and therefore more valuable. In an effort to simplify and optimize the way people spend their valuable free time, we have created an app, called “Yamsterdam”, to effectively find the nearest events and activities around them. The application allows a user to search for events by searching by keyboard input, GPS location, or voice input. The application communicates with Eventful, a web service that collects a large, diverse number of events occurring within a certain radius of a location. Our application finds the best events within a 25 mile radius, automatically filtering out events that are lacking in information. Users can find the title, description, date, and address location of any nearby event, and even save an event locally for future reference. The flexibility of inputs and simplicity of this application would greatly benefit the vast majority of society and improve the way people spend their leisure time.

Platform Justification:

Android was the team’s chosen development platform, as we wanted to implement Google API functionality. By using Android, we were able to fully harness powerful tools such as Google Maps and Google Voice Recognition. Google Maps allowed for the collection of GPS coordinates from the user’s device. Google Voice Recognition gave the user the choice of speaking a location to search, instead of typing a location or using GPS. Since our application is designed to function when the user is connected to the internet, Google APIs were used with minimal issues as there was a significant amount of useful documentation.

Key Features:

* Search by GPS location
  + The application allows users to harness Google Maps API to search for events that are close to their immediate location. (Requires user to turn on GPS)
* Search by Voice
  + The application implements Google Speech Recognition to search for events by speaking to the device. The result of the speech recognition is also displayed so the user is able to make minor corrections or validify the correctness of the recognition.
* Traditional Text Search
  + A third option of input, text search, is also available in case devices lack voice input or GPS service.
* Filtered and Processed event information
  + Before any data is displayed to the user, it is carefully filtered so that events without a sufficient description are not displayed. Additionally, since the Eventful API provides event description information as a String with HTML tags included, the HTML has been post-processed so that the description is displayed correctly.

Testing Methodologies:

Testing was performed using a series of unit tests and checks to meet user requirements. The testing of features was done by identifying what actions a user would want to perform, and the results that they would expect. For instance, a user would want to search for an event by GPS location, search through a series of events, find an event that interests them, and save it to a local “saved events” list stored on their device. Extensive testing was performed to ensure that the local database functioned and stored event data correctly. Thus, the in-app testing was mostly hands-on, where we found and corrected bugs pertaining to formatting, layout, and in-app functionality. We were able to discover a large number of bugs that were previously undetected, such as mismapped back buttons, formatting issues for event details, and poorly displayed error messages. User testing allowed us to make the necessary corrections to polish the application.

The second part of testing required ensuring that the application functioned on other Android devices, did not crash, and worked even if the application was paused, stopped, resumed, etc. Two devices, besides the provided Android tablet, were used to ensure that the application worked on other tablets, as well as small Android phones. Switching between applications also worked seamlessly, after a significant amount of debugging.

Usage:

The application is fairly intuitive to use, and thus usage is quite simple. An important thing to note is that the application can only be used to its full potential if GPS location and Google services are enabled on the device. These are required for the speech-to-text and GPS location search functionalities. Using one of the three provided search functionalities, the user can simply look up a location, which results in a list of events within a 25 mile radius to appear. The list of events is constantly refreshed and may display a different list of events when the Search button is pressed again. Any event on this list can be clicked to expose more information, such as the date, description, and location of the event. Additionally, the event can be saved to the user’s “saved events” list, by storing the event data as an object in the local database on the device. This page is accessed from the main search activity via a button on the bottom. Clicking this button opens the “saved events” list, where the user can select an event from this list to display it. Here, the user has an option to delete the event from their wishlist.

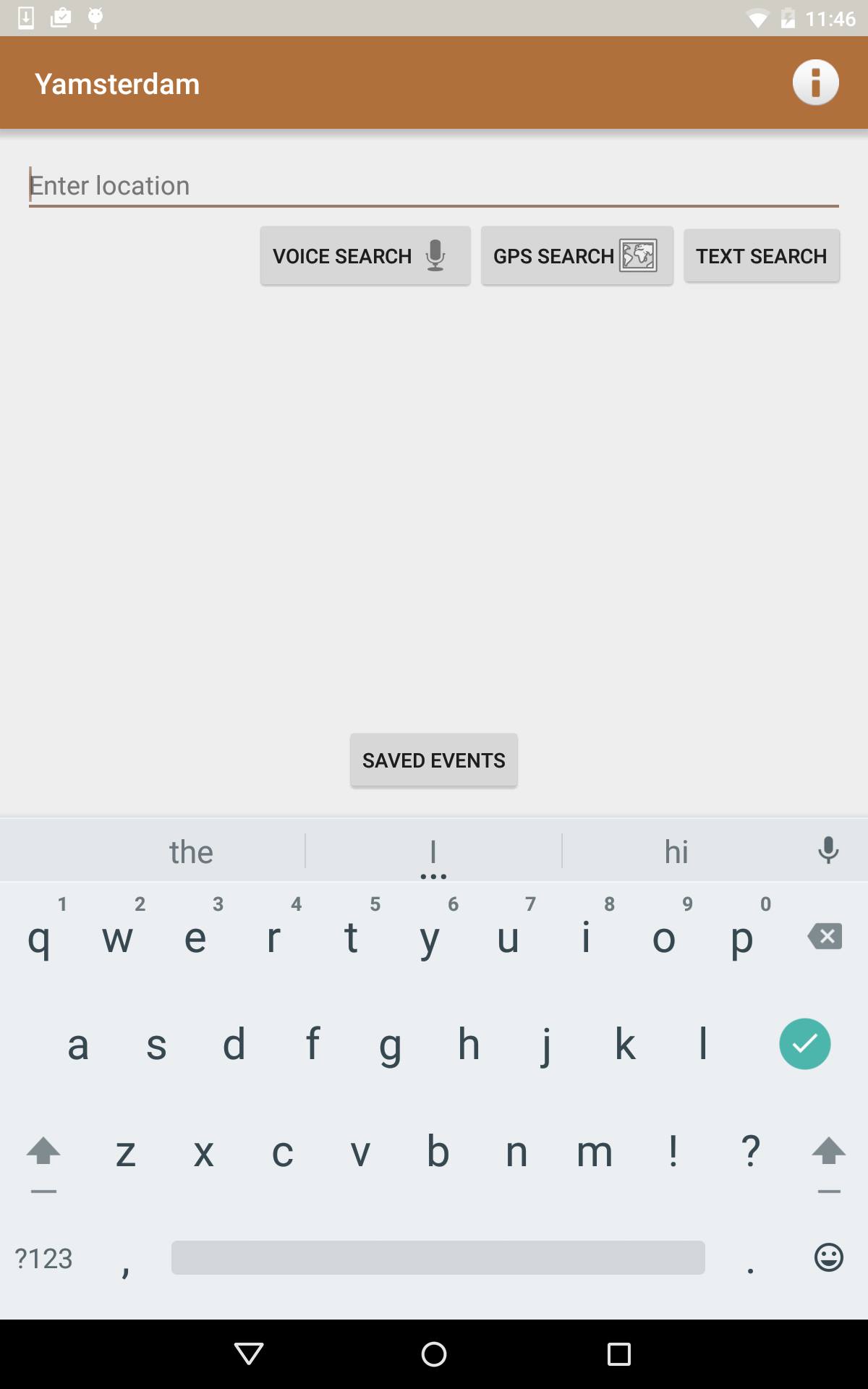
Lessons Learned:

This project was particularly useful in teaching both beginner and intermediate application development techniques, as we it forced us to consider cases where the application would be paused, stopped, resumed, etc. Additionally, the implementation of two hardware components was very helpful and immensely useful for future application development. The Google API is a very useful service that we were able to implement and understand through this project. We were quite floored at how simple it was to implement a seemingly difficult task like recording the user’s voice and decoding the sound to predict the corresponding words. In the future, we would definitely use Google services when developing with Android devices.

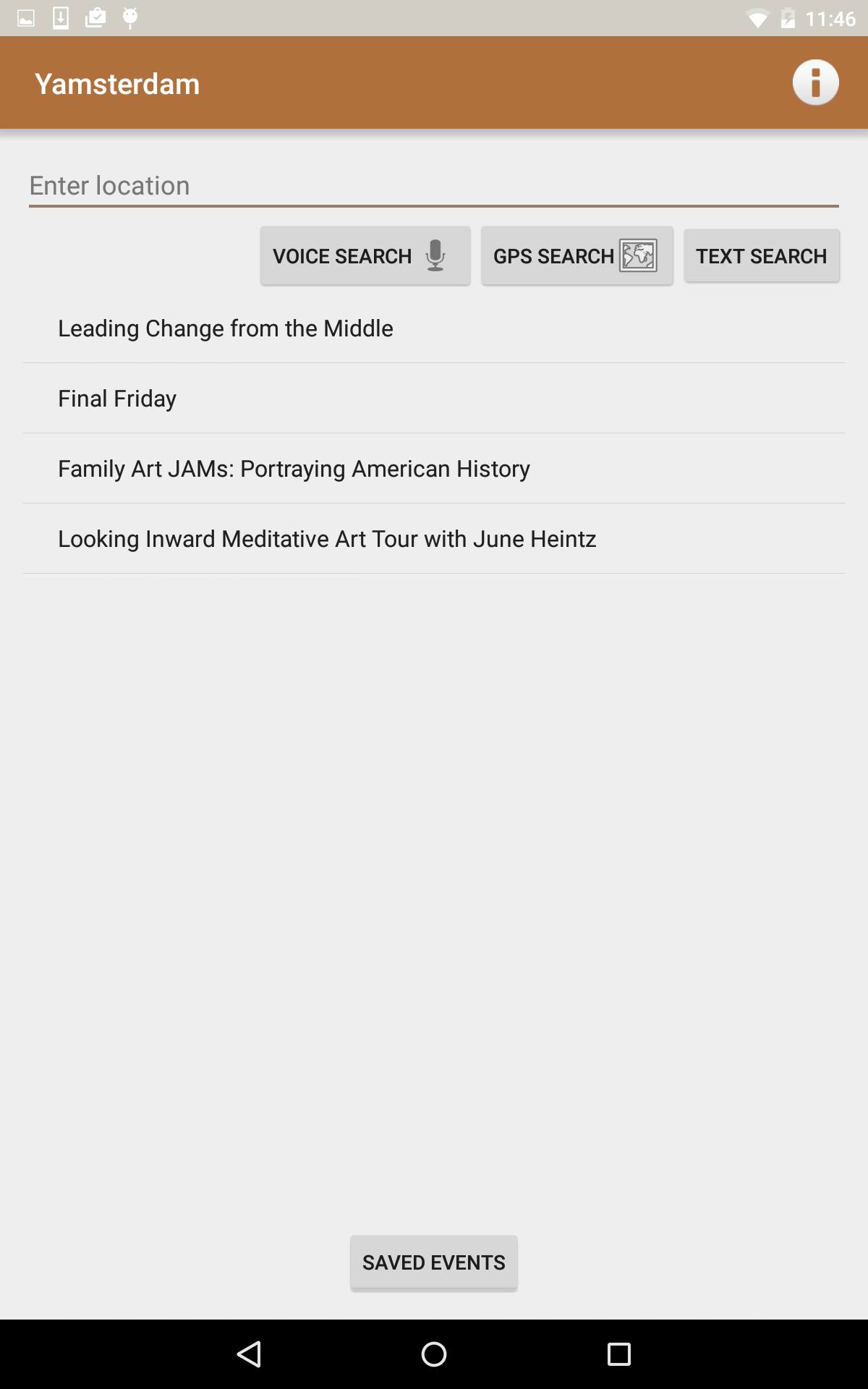
We were also able to learn how to use APIs such as Eventful to bring useful data to users, by parsing JSON data and removing HTML tags in strings to make them more readable. Before this project, we were mildly uncomfortable with the concept of using web services within our Android applications. Whereas before it seemed quite difficult, we learned that implementation is actually quite manageable, and that there is a significant amount of documentation available to ensure seamless integration. Additionally, several misconceptions about APIs were cleared, such as the notion that these web services lacked significant customization to tailor them to our needs. However, Eventful had a wide array of customizable attributes, including the size of the event search radius, among other things. It even allowed us to have two different methods of inputting location (GPS coordinates and text input).

Wireframe:

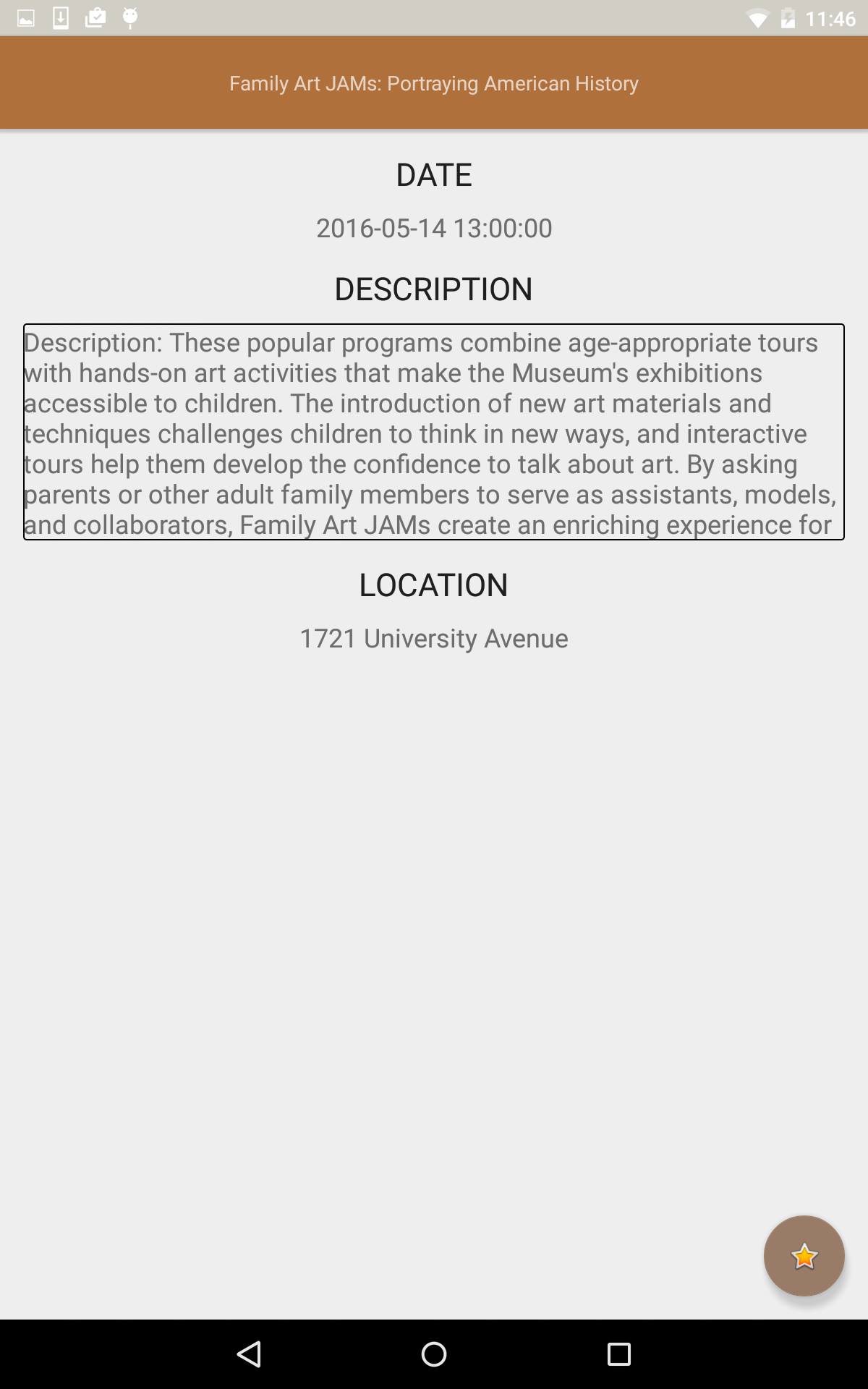
Home:



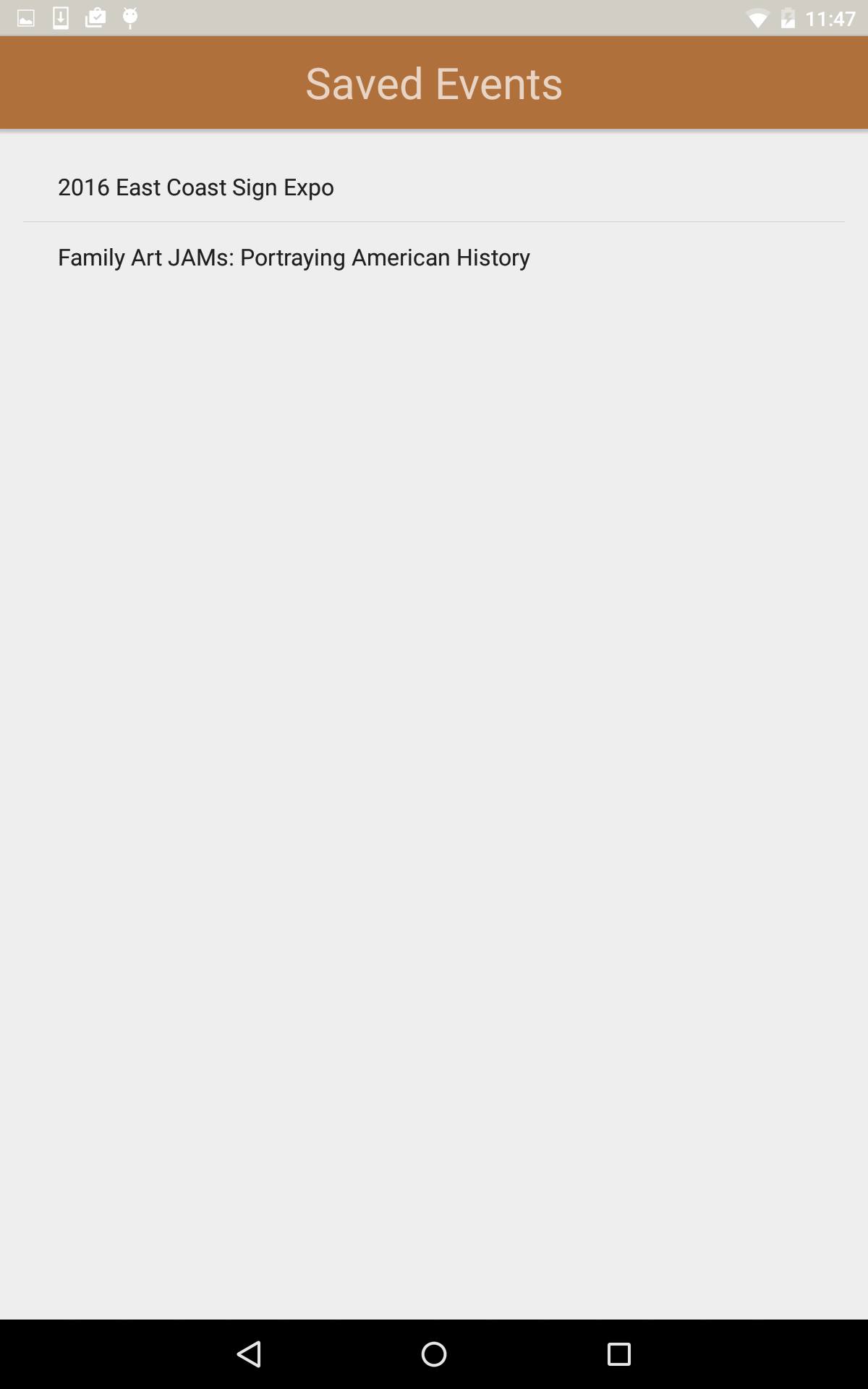
Search Results:



Info Screen:



Saved Events



App info Screen

