Abstract:

Evently is a mobile application built for Google’s [7] Android mobile OS [1] that enables users to search for and create events that are displayed on maps. The appearance of the application on the device resembles Google Maps [8]. The events are created by users exclusively on mobile devices. All events are publically available which means other users are able to browse the events posted, with a choice of using GPS data as input for their searches. They are able to filter the results with categories like “social”, “business”, “politics”, “sports”, “education”, “music” etc. When creating an event, the user has the option to enter an address or click on a point on the map when entering details about the event. When he enters an address the software tries to find the location on the map, in which case the user can either confirm that the location is correct or specify a new location manually. The devices’ GPS capabilities are important as the users are presented with events that are happening near them. Evently requires internet access in order to work.

Related Work:

Internet map usage was identified as a trend as early as the mid 1990s [9]. Since then the Internet has grown both in size and popularity. Nowadays maps can be found on mobile devices and in a number of different websites. Facebook [11] has an event function. Users can create an event and invite their friends. A map function is there to display addresses where available. A project done by students in the fall 2007 semester in the CIS350 course of the University of Pennsylvania utilized this Facebook functionality [6]. It maps Facebook events that the user or their friends are invited to. Nokia released “Here and Now” on the 21st of April, which is displays event and weather information. Not much else is known other than that the information will be provided by “3rd parties” and that it uses Cell Id positioning which is less accurate than GPS [4].
Some of the services mentioned above inspired the creation of Evently. The events on Facebook are user-generated, an idea adopted by Evently as well. Other sites like “What’s on When” [15] manage the content themselves. A mobile platform was chosen for easier, faster and more frequent access from users. When Google announced the Android OS in late 2007 it stressed on its location-based capabilities [13]. Naturally, a few applications have been developed that are similar to what we propose but ultimately target different functionality. Geocast [2] and Wikitude [14] make use of the mapping capabilities of Android but have nothing to do with event creating/finding. Geocast served as an initial inspiration for the GUI of Evently. Google Maps has an important role for this project. It provides the maps for the events to be displayed on and it assumes the task of mapping real addresses to coordinates\(^1\) for Evently to use.

**Structure of the application:**

When the user loads the Evently application, they have to login with their username and password or signup using their email. If the login is successful, they are presented with a Google Map which has the results of a “default” search after a GPS fix from the onboard GPS unit is obtained. The default search uses the data from the GPS to search for and return events that are taking place within the area of the map that is displayed on the screen. The “menu” button on the handheld provides options to start a new search, create a new event, view the events that are displayed on the screen (from a previous search) in a list or view the details of the selected event. In the search screen, the user has the option to search for events according to criteria that they select. Afterwards they are presented with a map with the results of their search represented as “pins” and labels on the map.

**High Level Technical Approach:**

The program consists of the client program running on a mobile Android device and a database running on a server. When a user creates an event, it is stored in the server’s MySQL database. When a user searches for events, the server performs the search and sends the results back to the client. The database includes all the information for an event

\(^1\) Anywhere the word “coordinates” is mentioned in this paper, geographical coordinates are implied unless explicitly stated as otherwise.
including name, time, date, location (coordinates and address), type of event, views (popularity). A signup/login system is implemented which requires the user to provide a user name, a password and an email address when signing up. Evently requires internet access in order to stream the map tiles. No information is stored on the device.

The most challenging technical difficulty was designing an algorithm that only returns a few results and not the whole database. This had to be done for bandwidth, storage and speed reasons. A user would not want to download megabytes or gigabytes of information when searching. This algorithm ranks the results according to popularity and past user preferences. The algorithm is described in detail in the next section. Other technical difficulties included interfacing with and using MySQL databases. The Android SDK does not include support for interaction with remote databases (only SQLite databases embedded in the application). A number of PHP scripts are used to handle the queries to the databases. The scripts run on the server and they are called from the client application on the Android devices. Data is passed to the scripts via URL strings and the client receives data via HTTP. Testing the GPS support was also challenging because the application is running in the Android OS emulator included with the SDK, which does not have a GPS receiver to work with. Instead of using the actual GPS service of the OS, which reports to be inactive and returns null coordinates, a point was chosen and hard-coded for the purposes of this project. Planning the project and learning how to use the tools that Google provides (the Android SDK and the Google Maps API) took a significant amount of time.

**The ranking algorithm:**

When a user performs a search, his device sends data to a PHP script on the database. The data include the user’s criteria and information about his location. The script sends a query to the MySQL database and gets back a list of all events matching the user’s criteria. This list is in no meaningful order. The script ranks the results and sends back the top 50 results. This way the amount of data is minimal (a few kilobytes) and the information is the most relevant. The results were also limited to 50 in order to avoid overloading the resources (memory, processor) of the devices. The 50-result maximum limitation should not be a problem for the user. Studies have shown that the majority of users do not look past the second or page of the results of the search page (40-60 results)[3].
Ranking mechanics:

- Events are assigned points. The more points an event gets the higher it ranks.

- Events get 3,000,000 points for a keyword match in their title and 1,000,000 points for a match in their description.

- Events get an extra point for every time a user has clicked “I like this” on them. This way, popularity of an event is taken into account when ranking.

- Events get a percentage of 1,000,000 based on the user’s preference. Preference is determined by how many times the user clicked “I like this” on the different types of events. For example, a user has clicked “I like this” on 3 “Music” events, 2 “Sports” events and 1 “Education” events. Music events in this case have 50% of the user’s preference and as a result will get 1,000,000 * 50 % = 500,000 extra points during ranking.

- The millionths of degrees on the geographical coordinate system are treated as if they were units of length (distance). Events lose a point for every millionth of a degree of their distance. The distance is calculated using the Pythagorean Theorem.

  The last step is the most important step of the algorithm since the user sees the events that are near his physical location first. Here is an example to demonstrate the effectiveness of the algorithm: There are two events with a match in title but not in the description and zero popularity and preference from the user (to make things simpler). One is located in Philadelphia, where the user is located, and the other is in New York. The one in New York will lose approximately 1,500,000 points. In this case the one in New York will rank lower than the one in Philadelphia. Negative scores are possible but that does not change anything because events that don’t match the user’s criteria are not included in the ranking process. Also worth mentioning is that no event is discarded because of a low score unless they fail to score among the top 50 events.
Development milestones and difficulty:

These were milestones set during the development process of Evently. The numbers in the parentheses show the amount of work/difficulty of each milestone.

- **Create program screens (4)**: Created the screens that show the map, and host the input fields for the search and create functions.

- **Create local database (3)**: Created a SQLite database that was stored on the phone and stored the events even when the program (or even the emulator) is not running. All the information about events was stored in the database in text except for the coordinates which were stored as integers. This database was removed when the database was moved to the server, but helped in the initial development of the client application.

- **Implement basic search (2)**: Implemented a search function that took the search terms and returned any event that had any one of those terms in its title. Results were unranked. All of the results were displayed with a red circle on the map. All events were first taken out of the database and put into a List. Then search was done and the coordinate data from the results were added to the list which was used to draw on the map. The mechanics of search changed completely, but this “basic” search was also helpful for developing the client.

- **“Find on map” feature (2)**: When creating an event the user is able to click on the map in order to pinpoint where exactly the event is taking place. This is done by handling screen “taps”. “Taps” have screen coordinate data which are converted to geographical
coordinates which are used to draw a mark on the map to give the user feedback regarding their input. These same coordinates are returned to the “create” screen.

- **“Clickable” search results (3):** The user is able to click on any of the events that are displayed on the map. The map automatically animates so that the event is in the middle of the screen and a popup window displays information for the event. “Hit testing” takes place on the main map screen. The screen is monitored for “taps”, and their screen coordinates are converted to geographical for testing. The program then iterates through the events that are displayed on the screen, checking to see if the “tap” was inside the icon marking the event.

- **Check input (2):** Implemented checks for appropriate input and give feedback to the user with popup menus. Checks look for missing text from input fields.

- **Pre-defined categories (1):** In the create screen changed the “category” input field to a choice of predefined categories using a “drop-down menu”.

- **Augment create screen (4):** Added more input fields that ask the user for important information like start and end time of the event. Also wrote the relevant PHP scripts.

- **Complete the search screen (5):** The user is able to enter search criteria other than just keywords in the title of an event. Search criteria mirrors the data entered when creating an event. Also wrote the relevant PHP scripts.

- **“Near me” search (2):** This function enables the user to find events that are happening near them. This type of search returns all the events happening within the limits of the screen (input is the GPS coordinates). These results of this search are automatically displayed when the application starts.

- **Rank search results (5):** (discussed extensively above)

- **Convert address to coordinates (3):** This function allows the user to enter a real address and the software tries to find it on the map. Then the user is presented with a few potential locations choosing one of them to confirm or manually override it. This feature is a complement to the “Find on map” feature. Using an address is faster but manually clicking on the map is more accurate. Google Maps is be used for the conversion. It is not guaranteed to work for all countries and regions.
- **Use widgets for input (2):** The Android SDK provides widgets for time and date input. Using these not only makes the program prettier but also makes it less prone to input errors and no check has to be made regarding the well-formedness of the input.

- **Implement GPS support (1):** Although GPS support is a vital part of the concept of the application, its presence is not noticeable in this project because the application is running in the emulator and there are no GPS data. For that reason a point was hard-coded as the “current location”. That point was chosen to be the Engineering building of the University of Pennsylvania.

- **Move database to server (5):** Implement databases on the server (more information in the “Resources Used” section) so that they are accessible from all devices in the world running Evently. Two databases are used: one to store events, one to store user data (logins, preferences). It also handles search.

- **Change map icon (2):** In the initial phase of the program, the icon that marks places on the map was a red circle. It was replaced by a better-looking “pin-like” icon similar to the one used in Google Maps. A blue version of the same icon is used to denote the user location (GPS)

- **Handle “back” key (1):** There is a “back” key on Android devices. It is handled to return to the previous screen, effectively cancelling any changes the user made in the screen that is showing

- **Add to calendar (NA):** (When a user is interested in some event they will have the ability to add that event to their device’s built-in calendar). Unfortunately this is not yet supported as of the most current version of the SDK (1.1_r1)

- **Optimizations (2):** Rewrote inefficient code as it was discovered. All the searching is done with the built-in functions of MySQL and the ranking takes place on the server to minimize data sent to the device and also minimize the application’s footprint on the device memory.

**Resources Used:**

As mentioned, the application was developed for the Android mobile OS. The Java programming language is the only one supported by the SDK. The SDK provides an emulator for a touch-screen phone in portrait mode. This is what the software targets because touch-
screen phones have become a trend since the iPhone launch [10]. The development took place on my Windows laptop using the Eclipse IDE [5] with the Android plug-in that comes with the SDK (SDK version 1.0_r1). The software only runs in the emulator for the purposes of the senior design project. Resources that were used are tutorials, examples and APIs of the Java programming language and of Android applications specifically. Documentation, tutorials and examples for MySQL and PHP were also referenced. A server connected to the internet was used to handle the database. The server is a LAMP setup (Linux-Apache-MySQL-PHP) running on Ubuntu Linux on a laptop.

Conclusion:

Eventually grew into a system that was greater than what was initially anticipated, especially on the server-side; PHP was not part of the original design but was needed to interface with the database. As development progressed, different aspects of the system needed more or less attention so focus would shift often. A few features that were planned never got implemented because of time constraints. For example have the option for “themes” on the client application, which would change the colors, was labeled a “low priority” and was left out. There were more features that were not implemented because they were not planned for and thus not included in the development schedule but would have been useful. Such features are the ability to edit or delete an event from the user that originally posted it, and also rank the event according to the date they take place (the further in the future the more point they would lose).

In retrospect, there are two steps that should have been taken to better manage development time: 1) Have a full specification for most aspects of the system and especially those of the data structures in the database; a well-thought out template that would be followed exactly. 2) Change the schedule to have the server-side ready before most of the client. This would have eliminated the need for the temporary database on the device that was ultimately removed.

Overall, the system is fully functional and conforms to the specifications that were set before development began. The original idea was expanded to include more features such as ranking and the system was adapted to make use of new tools and features that would be needed

   The Android OS was developed by Google. It is an open source project. This is the website that contains the Android SDK and the Android API documentation. The search function for the documentation works well, and the documentation itself is comprehensive. There are links for group discussion where help is given by other developers or from the Android developers.


   Geocast served as an initial inspiration for the GUI of Evently. It allows for sharing multimedia files which are displayed on a map. Its philosophy is similar to that of Evently but it targets different functionality.


   Skooiz describe themselves as “Pay Per Click Search Engine Marketing (SEM), Search Engine Optimisation (SEO), Email Marketing, E-Commerce” on their blog ([http://skooiz-en.blogspot.com/](http://skooiz-en.blogspot.com/)). The referenced document has information about search engine usage among French Canadian Usage.


   Nokia released “Here and Now” on the 21st of April, which is displays event and weather information. Not much else is known other than that the information will be provided by “3rd parties” and that it uses Cell Id positioning which is less accurate than GPS.

The home page of the Eclipse IDE, the officially supported IDE for Android development. Eclipse is open source software.


This is a project done by students in the fall of 2007 in CIS350. It maps Facebook events that you or your friends are invited to. This website is limited to Facebook users and also limited to the University of Pennsylvania network. Facebook Events Locator inspired me to create something on a grander scale.


Developer of the Android OS.


Google’s mapping service. It will be this service’s maps that Evently will use. Google Maps will also be responsible for converting real addresses to coordinate data for use in the application.


This article was written by Michael P. Peterson, a student in the Department of Geography / Geology of the University of Nebraska at Omaha. He gives statistics about map use on the internet and also statistics about general internet use. The article does not mention when it was written but I estimate late 1990’s because his latest statistics and references are from the year 1997. Even though the article is old it shows that map usage on the Internet is quite old as a trend.

   This is an article in a blog that shows that touch-screen phones are becoming the new trend in mobile phones. This is the reason that Evently is targeted to a touch-screen Android phone.


   Facebook is a social networking site. Among its services is an event function. Users can create an event and invited their friends. A map function is there to display addresses where available, but the user cannot point on the map where exactly the event will be taking place. The events are semi-private; you can only view events you or your friends were invited. Also there is no (easy) way of accessing these events on your mobile device.


   This article provides a short description of binary search and a sample implementation in C/C++. It has the tone of a tutorial and people can ask questions about it on the bottom of the page.


   Written by Google, it gives an overview of the features and capabilities of Android as a platform. It provides information about the application framework, the libraries, the runtime and the Linux kernel of Android. This page serves the purpose
of attracting developers to program for Android and is not particularly useful once you start developing.


This application gives users the ability to share interesting locations. Like Geocast, its philosophy is similar to that of Evently but it targets different functionality.


This website has information about events on a global scale. The content in this website is not user generated; the company manages it.


This is a tutorial that helped me with coding. It highlights a few useful functions for manipulating maps in Android. Its author is a developer working for Microsoft.


This is a tutorial that helped me with coding. It has sample code for drawing on maps in Android. There are reader comments on the bottom of the page which include questions.

This is a tutorial that helped me with coding. It has sample code for testing if the user has clicked on something on the map. It also includes a bitmap of the pin icon used in Google Maps which I plan to use. There are reader comments on the bottom of the page which include questions.


This is a tutorial that helped me with coding. There are reader comments on the bottom of the page which include questions.


This is a tutorial that will help me with coding. It uses the Geocoder class that I will need to use to convert addresses to coordinate data. It includes sample code and comments and questions at the bottom of the page.

21. The following sites helped me with coding. They are tutorials and examples mostly from blogs. They helped Android programming but also PHP and MySQL.

http://www.linuxdevices.com/articles/AT6247038002.html

http://www.helloandroid.com/node/159
http://www.codediesel.com/php/linked-list-in-php/
http://groups.google.com/group/android-developers/browse_thread/thread/db3eee0abe7aafe4