Utilizing Multiple Location-Based Service APIs for an All-In-One Mobile Application "Check-In" Solution

David Ivins
Drexel University
3141 Chestnut Street
Philadelphia, PA 19104 USA
+1 856 371 0204
davidivins@gmail.com

ABSTRACT
Location-based services are the latest in popular Internet and mobile phone technologies. Numerous location-based services such as Foursquare, Gowalla, Yelp, Brightkite, SCVNGR, and Facebook Places are being created everyday. As such, numerous mobile phone applications are being developed to allow users to utilize these location-based services from anywhere on the planet. Though these services and applications are very useful, a user can actually be overwhelmed when attempting to use more than one location-based service at a time. Using more than one service at a time often involves using more than one application for performing location-based actions, such as "check-ins". CheckIn4Me is an attempt to consolidate "checking in" on numerous location-based services to one application by utilizing the services' public APIs and the Android Software Development Kit. For this study, a proof-of-concept application was implemented using the Foursquare and Gowalla APIs. It was then tested to show the time saved by using one application to "check-in" on multiple services and to show the overall satisfaction with the user experience.

Keywords
Location-based, Mobile, Smartphone, Social, API, Android, Check-in, Computing, Architecture

INTRODUCTION
In 2007, [1] stated "a mobile device is ideal for sending and receiving location-based postings, since it is always with the user. When GPS enabled mobile devices will become broadly available, the amount of location-based information will grow fast and location-based posting will become more widely used." Coincidentally in 2007, Apple introduced the iPhone. Ever since, smartphone technology has reached beyond the business world and into the mainstream to become a dominating force in popular tech culture.

Prior to Apple's entrance into the market, only businessmen and technology enthusiasts primarily used smartphones. Now, those ranging from high school students to senior citizens are carrying smartphones. In addition to increase in popularity, there has also been a dramatic increase in the amount of functionality offered by these devices. Many years ago, smartphones offered only basic functionality, such as web, email, and document management. Now they are coming equipped with global positioning systems, accelerometers, and gyroscopes as standard features. Two of the more popular operating systems included on phones of this nature are Apple's iOS and Google's Android. Both of these operating systems offer a software development kit and a vast amount of documentation. Given the new phone hardware and the accessibility of the SDKs, a wealth of new and interesting mobile applications are being developed for these platforms [2]. These interesting applications cover areas such as location-based services, augmented reality, and interactive gaming [3].

Location-based services, in particular provide a number of useful features to users, including social networking, mapping, organizing, planning, searching, and sharing of information. [4] Showed that over 70 percent of the conversations they studied contained information about location or activities. People naturally want to share information about their location and activities, and location-based services provide a convenient and easy method for doing so.

Furthermore, through public APIs, these services allow for third party developers to build upon these services as well as extend them. This paper evaluates an application built to utilize multiple location-based service APIs in order to provide an all-in-one easy-to-use "check-in" solution for an Android-based mobile phone. [5] States "Location-based computing depends heavily on the technologies on which it is deployed and how it is applied. In order to achieve effective, pervasive deployment of location technologies, the supporting software must run on a variety of platforms including laptops, PDAs, and mobile phones." Though this is true,

we argue that in order to achieve "effective, pervasive deployment" of location-based services, there must also be a common piece of software for interacting with these services on each device. Currently that does not exist due to each service and their respective applications being segmented from other services. CheckIn4Me does not create another segmented service or piece of software. We take the same approach as [6] in that "rather than seeking to replace the current practice with technology, the aim should be to enhance and support it, in order to make location-based services which are relevant, innovative and perhaps also fun to use." We also aim for the "perceptibility, flexibility or low effort" mentioned in interviews conducted by [7]. Therefore, through use of the public APIs, we hope to achieve a common easy-to-use software interface for location-base services on Android devices.

BACKGROUND

To date, numerous Location-based mobile applications exist for the Android and iOS platforms. These applications serve many different purposes. For instance, Yelp is primarily a review site for businesses. Where as SCVNGR is geared towards scavenger hunts, location challenges, and adventuring. Foursquare, Gowalla, and Brightkite lean more towards the social networking side of applications. With these three services, users maintain friendships and interact with their friends through messaging or multimedia.

Although these products serve different primary purposes, they also share a lot of location functionality in common. For instance, many provide maps of where users are located and what is around them. Another feature in common is the gaming aspect of location-based applications. All of these services achieve a gaming aspect by allowing users to collect things. Many reward users with virtual prizes for going to certain places or doing certain things. [8] Notes "tying objects to location imbues them with particular value because one has to go to the location to retrieve the object. The collected object becomes proof of the visit to the location; a kind of digital souvenir." In addition to this, all of the services allow users to "check-in" at their current location. This gives users the ability to collect all of the places they have ever been along with all of the digital rewards they have ever won. A study performed by [9] showed that in a simple location-based game, there was "evidence of a very high fun factor, independent of age, gender, playing conditions, and inclination towards sports and physical exercise." Thus, the gaming and collecting aspects of location-based services are very powerful ways to increase usage and interest among each service.

Though it may be called something different across applications, the idea of "checking in" is the same, and it gives the user the ability to mark their current location at the current time. They can then notify their friends of their location or allow others to see their current location. To some mobile phone users, this seems like the ultimate invasion of privacy; the ability of anyone to track your personal movements throughout the day does not appeal to everyone. However, to many, the ability to "check-in" has many useful benefits. It allows users to let their friends know that they are currently at a bar or night club so others can come join them. This can take a lot of the phone calls or text messages out of planning and organizing a night out [10]. "Checking in" at certain locations can provide useful information to the user as well. If a user is on vacation and "checks in" at a nearby restaurant, patrons that have previously been to the restaurant can provide information on the best dish or what food to avoid. The vacationer can even use it to decide where the best meal is prior to "checking-in". Advertising can also be made useful, rather than annoying, via location-based information. It has been shown that users actually respond well to location-based ads [2], similar to the way targeted ads on the web work better than random ones. Foursquare, for example, has a built-in specials capability for businesses with locations on Foursquare. A manager can offer discounts to the patron with the most "check-ins" at the business. Finally, the "check-in" ability is just fun, which is what appeals to many users. It let's the user keep track of where they have been, earn prizes, and keep in touch and play games with friends.

However, as useful and fun as "checking in" is on all of these location-based applications, there is still a major problem; there are too many location-based applications to "check-in" with. Unlike websites such as Google, the king of search, Facebook, the king of social interaction, or Wikipedia, the king of information, there has yet to be an application crowned king of the location-based mobile applications. Foursquare is arguably the most popular, however, since this is a fast growing and very popular market, many big names are jumping into area as well. Facebook, for instance, just launched its "Places" functionality over the summer, allowing users to "check-in" while posting their statuses from their iPhones. With so many popular new players and established big names in the mix, there are currently too many applications to "check-in" with and no way of deciding which is going to be the most popular in the future. This essentially forces users to choose between a select few of the location services since nobody wants to take out their phone and "check-in" on 10+ different applications every time they go to a new location. It would be far too time consuming. Unfortunately, it is still difficult to determine which service will ultimately win the popularity race, and many people have friends spread across multiple services. CheckIn4Me aims to provide a solution to the problem of "checking in" over multiple applications and services. CheckIn4Me will allow users to connect their location-based service accounts to CheckIn4Me, via the services' own APIs, and then "check-in" once from within CheckIn4Me across all of their connected services. This

provides a quick way to "check-in" while not limiting the services with which the users can share their information.

SYSTEM

CheckIn4Me utilizes a number of technologies to achieve its goal as an application. First and foremost, CheckIn4Me runs on the Android mobile operating system. Thus, it uses the Android software development kit provided by Google. Android is a Linux-based operating system. The software development kit provides all the tools a user would need to build an Android application. It includes libraries, a debugger, and an emulator, all of which integrate into the Eclipse IDE via the ADT plug-in. All applications, including CheckIn4Me, are written in Java and run on the Dalvik virtual machine.

For connecting services to the CheckIn4Me, the OAuth protocol was used. OAuth was chosen because most of the APIs provided by location-based services implement some version of OAuth for connecting to their services. OAuth allows users of CheckIn4Me to authorize the use of API calls to access private user data without exchanging any username or password information with CheckIn4Me itself. Multiple versions of OAuth are used in CheckIn4Me. For example, Foursquare implements OAuth version 1.0, whereas Gowalla implements a draft version of the upcoming OAuth version 2.0. Though very different in the details, they generally work as follows: The application requests a token from the service implementing OAuth. The application then sends the user to the service's authorization website for OAuth with the token and a redirect callback. When the user authorizes the application to make API calls on the user's behalf, the service redirects the user back to the application using the callback and returns the token. The application then exchanges the token for an access token. Once the application has this token, it provides it in every API call to the service in order to retrieve and publish user data.

For interacting with multiple location-based services, CheckIn4Me contains implementations of each service's public API. Each API request is performed over HTTP using the OAuth protocol. Each request results in a JSON encoded response that can be easily parsed and interpreted by the receiver. This allows the application to retrieve locations from the services based on the Longitude and Latitude coordinates provided by the Android operating system (which will gather data from a GPS provider or a network provider) and allows the application to "check-in" for a user by sending the service the user's desired "check-in" location. A typical sequence of events is shown in figure 1 on the following page. This shows the connection of a service, the request for nearby places, the request for location details, and finally, the request to "check-in".

The back-end of the system attempts to implement all services as a generic Service interface. The interface provides a generic OAuthConnector interface and a generic APIAdapter interface. The OAuthConnector interface provides an interface for connecting to a service via OAuth. The interface is implemented differently for each service, so how the connection is made is service-dependant. However, the front-end of the system interacts with all OAuthConnectors in the same manner. The implementation of the APIAdapter is also service-dependant. However, each APIAdapter follows the given interface for common API calls such as "checkIn" and "getLocations". Figure 2 on the following page shows an example of the service interfaces and an example of the OAuthConnector and APIAdapter interfaces under the Foursquare service.

The front-end of the system provides a sleek interface for "checking in" on Foursquare and/or Gowalla. When loaded, the application starts a thread for each service connected and requests the nearby places from each. The user is then presented with the list of nearby places. Places that are on both Foursquare and Gowalla are merged into one list item. Each list item shows an icon below its name for each service it was found on. When a user selects a place, the application loads the location details, which includes the location name, address, map of the surrounding area with stars for the user location and place location, and a list of services it is available on. The user checks which services to "check-in" on (by default, all connected services are checked) and presses the "check-in" button. The application creates a thread for each checked service and requests a "check-in" at the current location for each. It then presents a success or failure dialog to the user and returns to the nearby places screen. Figure 3 shows example screenshots of the nearby places list and the location details screen.

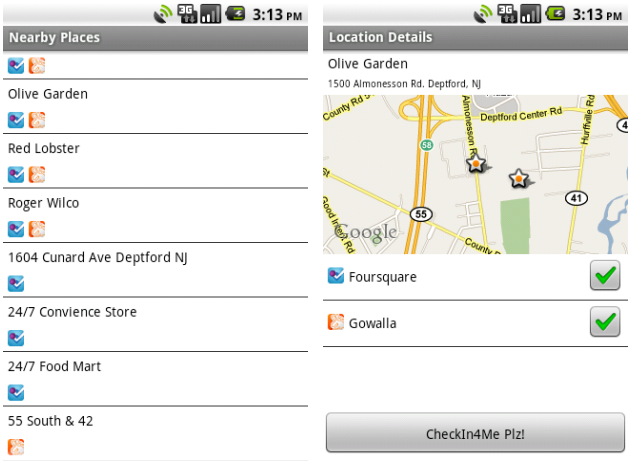

Figure 3: Screenshots of CheckIn4Me

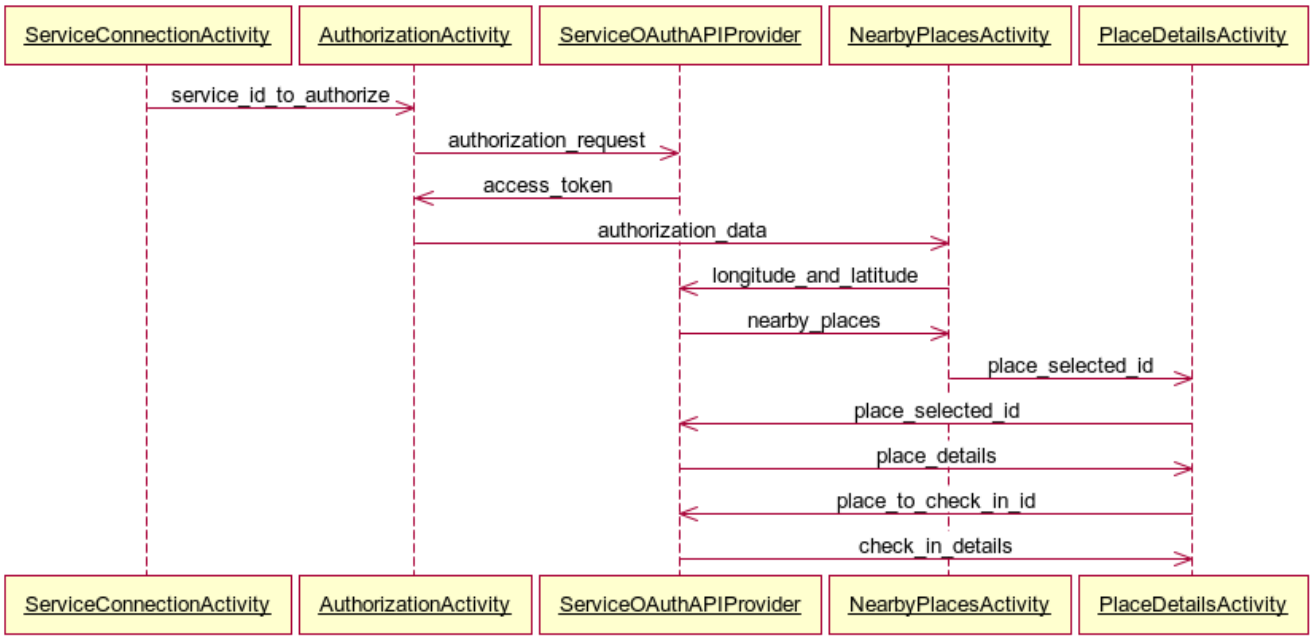

Figure 1: Typical Sequence Diagram of CheckIn4Me

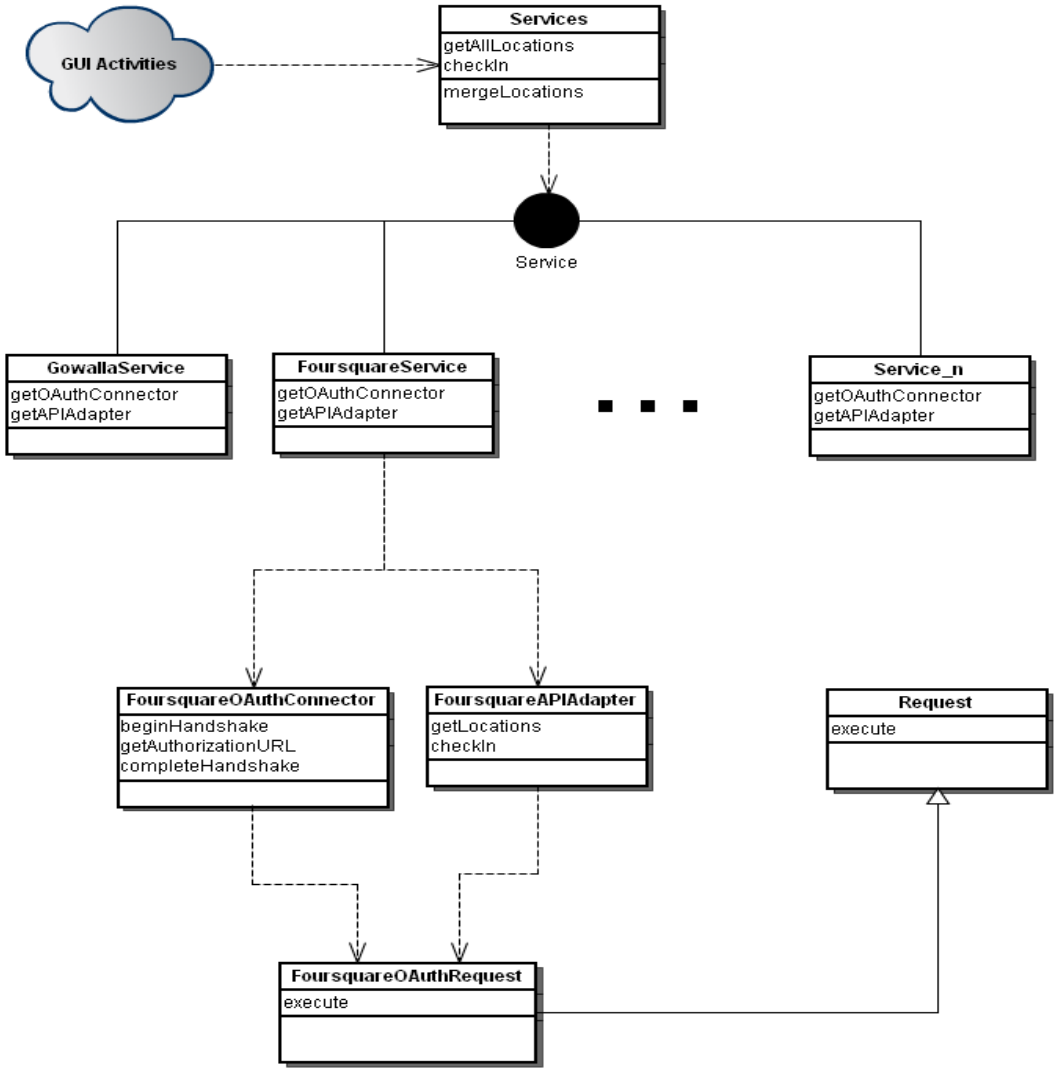

Figure 2: UML of Foursquare backend of CheckIn4Me

EVALUATION

CheckIn4Me was tested using an informal group study of four participants. The four participants were familiar with Foursquare and aware of other location-based services. The group also had android-based smartphones. The group was asked to create accounts with Foursquare and Gowalla if they did not already have accounts with those services (Everyone had Foursquare accounts already, not everyone had Gowalla accounts). After doing this, users were asked to do timed runs utilizing the CheckIn4Me, Foursquare, and Gowalla Android applications. The first timed run was the speed of logging in or connecting services. The users were asked to login on both the Foursquare and Gowalla applications as fast as possible. The time was recorded from the launch of the first application to the completion of second application's login. The users were then asked to connect both Foursquare and Gowalla's services to the CheckIn4Me application. The time was recorded from the launch of CheckIn4Me application to the successful connection of the second service. The time for connecting two services on CheckIn4Me was actually slower (roughly 1 minute and 10 seconds) than the time for connecting to both Foursquare and Gowalla applications (roughly 1 minute and 45 seconds) due to the fact that CheckIn4Me takes the user to the "Nearby Places" list upon successful connection of a service. By doing this, the application attempts to retrieve the phone's GPS coordinates and then makes API requests to all connected services. While doing this, the application is locked-up while a progress dialog is displayed. This is not ideal behavior for connecting services, thus, it will be fixed in the next version of CheckIn4Me. Also slowing the process was the fact that OAuth requires transferring control of the application from the application to the browser. Thus, for two screens (login and authorization), the user is crippled by the speed of OAuth and the browser on their phone, which is beyond the control of CheckIn4Me.

After the services were connected to CheckIn4Me and the Foursquare and Gowalla applications were logged in on, a second timed test was performed. Users were asked to "check-in" at their current location on both Foursquare and Gowalla. They were first asked to "check-in" using the location-based services' applications. The test was timed from the start of the first application to the successful "check-in" on the second application. They were then asked to "check-in" on both services via CheckIn4Me. They were timed from the start of CheckIn4Me to the successful "check-in" on CheckIn4Me. The time saved using CheckIn4Me (roughly 35 seconds) was significant when compared to "checking in" on both the Foursquare and Gowalla applications (roughly 1 minute). This was due to the fact that users had to quit from one application (ie: Foursquare) and load a second one (ie: Gowalla) in order to "check-in" on both services. Doing this caused the GPS coordinates to be requested twice, as well as additional unnecessary application loading times for location lists and location details. CheckIn4Me's cross-service merging of locations and utilization of threaded API requests to the connected services only required one GPS request and resulted in a much quicker dual listing and dual "check-in" for Foursquare and Gowalla.

After time trials, users were also asked to provide feedback on usability and functionality of the applications. There were a number of bugs found during this stage that were fixed prior to the application demonstration. There was a state data issue where locations attached to only one service were showing up with two services listed in their location details. This was due to state data not being cleared on a reload of the application. Another issue was with run away threads killing battery life. Initially, the application would try to get GPS coordinates and never timeout. If the user was out of GPS range, this was a problem, because closing the application did not stop the thread. This will be fixed in the next version of CheckIn4Me.

DISCUSSION

Despite the slower service connection of CheckIn4Me (which will be minimized or fixed by not returning users to the "Nearby Places" list upon successful service connection), it seems like users will save a lot of time with long-term use of the application. This is due to the fact that users only need to connect services every time the application's private data is cleared. This only happens when a user specifically clears it in the settings, or when the user installs the application for the first time. This means, more than likely, the user will only be connecting a service once for the lifespan of the application. However, users will be checking-in numerous times over the lifespan of the application, which is where the real time is saved.

Future work for CheckIn4Me includes implementing some of the major features existing on the services that are still missing from CheckIn4Me. These features include posting a message when "checking in", posting a photo from a "check-in" location, as well as searching for locations in the nearby area. Future work also includes incorporating other services. For the purpose of this project and paper, CheckIn4Me only implemented Foursquare and Gowalla API functionality. The idea is to incorporate as many location-based services as possible, which includes Facebook Places, Brightkite, Yelp, SCVNGR, etc.

REFERENCES

1. Lemmela, S., Korhonen, H. Finding Communication Hot Spots of Location-Based Posting. *CHI 2007.* (April – May, 2007) 2549-2554.
2. Rashid, O., Coulton, P., Edwards, R. Providing Location Based Information/Advertising for Existing Mobile Phone Users. *Springer-Verlag London Limited.* (November 2006) 3-10.

3. Rashid, O., Mullins, I., Coulton, P., Edwards, R. Extending Cyber Space: Location Based Games Using Cellular Phones. *ACM Computers in Entertainment 4, 1* (January 2006) 1-18.
4. Bently, F., Metcalf, C. Location and Activity Sharing in Everyday Mobile Communication. *CHI 2008.* (April 2008) 2453-2462.
5. Sohn, T., Griswold, W., Scott, J., LaMarca, A., Chawathe, Y., Smith, I., Chen, M. Experiences with Place Lab: An Open Source Toolkit for Location-Aware Computing. *ICSE'06* (May 2006) 462-471.
6. Weilenmann, A., Leuchovius, P. "I'm waiting where we met last time": Exploring Everyday Positioning Practices to Inform Design. *NordiCHI '04.* (October 2004) 33-42.
7. Jedrzejczyk, L., Price, B., Bandara, A., Nuseibeh, B. On the Impact of Real-Time Feedback on Users' Behaviour in Mobile Location-Sharing Applications. *Symposium On Usable Privacy and Security 2010.* (July 2010).
8. O'Hara, K., Kindberg, T., Glancy, M., Baptista, L., Sukumaran, B., Kahana, G., Rowbotham, J. Social Practices in Location-Based Collecting. *CHI 2007.* (April – May 2007) 1225-1234.
9. Misund, G., Holone, H., Karlsen, J., Tolsby, H. Chase and Catch – Simple as That? Old-fashioned Fun With Location-Aware Mobile Phones. *ACM* (October 2009) 73-80.
10. Schmidt, A., Holleis, P., Hakkila, J., Rukzio, E., Atterer, R. Mobile Phones as Tool to Increase Communication and Location Awareness of Users. *Mobility 06* (October 2006) 1-7.