

# Grab a Fika - Facilitating face to face meetings in the mobile age

Anton Malmquist  
Chalmers University of Technology  
antonm@student.chalmers.se

Olof Bjerke  
Chalmers University of Technology  
bolof@student.chalmers.se

Kevin Orende  
Chalmers University of Technology  
orende@student.chalmers.se

Rostislav Raykov  
Chalmers University of Technology  
z@zbrox.org

## ABSTRACT

This paper describes the rationale, design process and evaluation of the mobile application “Grab a Fika.” This application breaks the norm of effectiveness in regard of mobile applications, since it provides a method of meeting up and navigating that will not always give the easiest or most efficient means of transportation, but provides a novel view on how to find your friends. This is achieved by calculating and finding a café located in the geographical midpoint between the involved users. After doing this the application presents the user with simply an arrow pointing to the café and the distance to it, something that is not more efficient than agreeing on a time and place beforehand, but instead more interesting.

Most of the applications in today’s market have the capacity to deeply engage the user. Despite their inherent entertaining and engaging qualities, these mobile applications detract from interpersonal relationships. Everything is through a screen or encloses us within a mobile bubble, for example texting, playing games, Facebook etc.

During the project a prototype application was developed and user tested. The user testing showed that the app was usable and providing a new way to think about meeting up, but it was difficult to prove whether the app would be usable on a larger scale in the limited scope of the project.

### Keywords

*Mobile Computing, Navigation, Effectiveness, Mobile bubble*

## 1 BACKGROUND

In 1876, the first phone was introduced to the masses[1]. A massive event in that, for the first time people could orally communicate with others from a distance. Presently, those phones have undergone an evolution and are now more mobile and ubiquitous than ever. We are always connected, and as a result life has become easier and the world feels smaller[2]. The pace and efficiency of life has increased while also allowing more flexibility in business and professional life as well as family and personal life.

Paper presented at SIDER’14

Royal Institute of Technology, KTH, Stockholm, Sweden

Copyright held with the author(s)

The way we relate interpersonally has also changed. Since we rely heavily on our mobile devices for communication and entertainment, focus has shifted from the people presently around us to whatever is happening on our screens. In a sense we have been trapped in a “mobile bubble” where we are very aware of what’s happening in the virtual world or within the bubble but totally oblivious to events or people right around us. To combat this, we use the “bursting the mobile bubble” approach. This approach aims at designing collaborative, co-located experiences using mobile devices[3]. Therefore the purpose of the grab-a-fika application, is to burst the mobile bubble by encouraging and facilitating social meetings; more specifically to grab a fika.

In Sweden, fika is considered a social institution. The word fika could be defined in two ways. As a noun, it means a drink or coffee; As a verb, to have a break with ones friends, family, a date or colleagues. With the grab a fika application, we strive to make the planning and meeting of “getting a fika” more fun and easy with the use of mobile devices.

## 2 CONCEPT

The concept is of an application which would help a group of users to find a café that is equally close to all of the users in the group, and then help all users to get there. This is done by inviting users via email, sms or iMessage. After invites has been sent and the invitees have responded to the invites, the application shows an arrow and a distance. The arrow points towards a café close to the group’s geographical midpoint, as seen in figure 1, so that the group can meet up there and have a coffee. The distance shown is the distance to the café given in meters or feet based on the users locale.

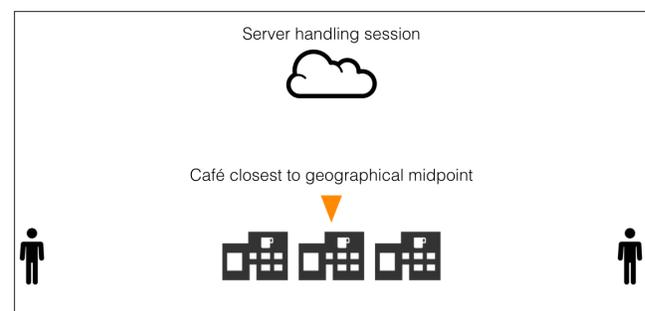


fig 1: Conceptual overview

To start up a new session with the application, one user with the app's help creates an invitation message which is then sent to the other users. When a message is received the application can be opened directly from the message to accept the invitation. If the invitation has not been accepted in a time specified by the initial user, the invitation will expire. If, however, the invitation is accepted the server will start a new session and find a café that is close to the geographical midpoint of all users that accepted the invitation and are now part of the session.

When navigation has started, the interface shows a simple arrow and a distance to the location due to multiple reasons. Firstly, just showing the arrow makes the application into more of a game than a utility. The user actually has to be creative in how to get to the point, but they can also choose their own preferred mode of transportation. This also contributes to making the application engaging in another way, since its goal isn't mainly to find the quickest and easiest way of getting a coffee with friends. That will always be deciding on a spot and a time together. This application instead helps you explore the city, by finding new ways to get to places, but also finding new places to get to.

Also the application can find new coffee places as the geographical midpoint changes due to users moving around. Every time a location update is sent to the server, the geographical midpoint is recalculated and a the café closest to the new midpoint is found. This can only be made to work by not showing the café name to the user, because otherwise the interface would get confusing, showing different destinations during different times. It also introduces a problem with the arrow changing direction during navigation, although this problem will mainly appear when one user is moving in a small circle around the others, something that is unlikely to happen.

This also means that the user actually has to use the app itself to use its functions, showing a name of a café would let the user copy that name and paste it into another navigation application to actually get turn-by-turn directions to the destination.

Almost coincidentally this also solves the problem of two cafés being equally close to the midpoint, the users will simply choose the café themselves when they meet up. When they do meet up the navigation will stop and the screen will instead show a list of all users in the session, and the distance travelled by each of them. This is shown to encourage people to actually move, since the navigation model permits one user to stay in one place and wait for all others to come to them, due to the midpoint being recalculated. Combined with the fact that the navigational model is not really transparent to the user, this should ensure that users are using the application in the intended way.

## 2.1 PROTOTYPE

The developed prototype consists of two separate applications, one server application and one iPhone application that acts like a client. The server's main purpose is to keep track of sessions, midpoints and users connected to that session, while the iPhone application is the user's interface to the system.

### 2.1.1 Client

The client application starts with a screen containing a single

button, saying "Send invitation" (shown in figure 2). An invitation is a message containing a URL which can be sent via e-mail, SMS or iMessage. Technically the user can copy the URL and send it via any service that lets them send text. This URL is using iOS's support for custom URL schemes [4] and is of the form "grabafika://[session\_id]". iOS's URL scheme support means that since the app is registered for handling "grabafika://\*" URLs the request will be handled by the prototype application. Thus the session ID can be extracted from the request. This also means that if the invitee doesn't have the application installed it will open the application's page in the iOS App Store. As the user doesn't need an account in order to use the app, it also conforms to the iOS Human Interface Guidelines [5].



fig 2: Application start view

The session ID is created by the server on request and is simply a unique token consisting of around seven characters. In the prototype, only two users can use the app to find a suitable location. Therefore every session ID can only be associated with two user ID's. When the accept URL is pressed on the invitee's phone a join request is sent to the server. The server will send a push notification to the inviter to tell them that their invitee has joined. Both the invitee and the inviter will then have the navigation started. This means showing an arrow and a distance, using the GPS location and compass heading of the phone.

Actually sending the invitation is done by accessing the iOS address book. Currently there is no real support for inviting people that are not in the user's address book. The application simply shows a pane with the user's contacts. When the user selects a contact the available means of contacting that user are displayed. Depending on what kind of messaging mode is selected an e-mail or text message editing pane is shown. A standard message "Hi! Want to grab a fika with me? grabafika://[session\_id]" is prefilled in the message field. The user can then edit the message as desired and send the message to the invitee.

Regularly the iOS application will send an “update” request to the server with a new location, telling the server to recalculate the midpoint using the new location and possibly find a new café (depending on how far the midpoint has moved). The response of this will be a target location, so the arrow will start pointing there instead (as in figure 3). These changes will most of the time, bar some edge cases, be small.

Carrier 10:25 AM



Distance to destination  
450m

Waiting for venue name

Cancel session

fig 3: Application navigation view

## 2.1.2 SERVER

The servers responsibility lies mainly in keeping the clients in contact with each other. Every session ID is associated with two users and a target location based on the users’ locations. The main resources available from the server is create, join, and update.

### 2.1.2.1 Create

To create a session on the server, the client sends an HTTP POST-request containing a user ID, and a location. The server then creates a unique ID for the session and sends that to the client in the HTTP response. This session ID is used for joining and updating, together with the user ID. The user ID is also needed to be able to send a push notification to the user. When another user joins the session, a push notification is sent to the initial user to indicate that the navigation can be started.

### 2.1.2.2 Join

When an invitee want’s to join a session, the client sends an HTTP POST-request containing their user ID, their location, and the ID of the session they want to join. In response, a target location is sent, so that the client will know where they’re going. This method also sends the aforementioned push notification.

### 2.1.2.3 Update

The update request serves two purposes. Firstly, when the invitee has accepted the invitation and joined the session, they know where the target location is. But since the inviter only has created the

session, and at that moment in time the session was only consisting of a single location so no target could be calculated, they don’t know the target location. So when the invitation has been accepted, the client of the inviter sends an HTTP POST-request to get an initial target location, and at the same time updates their location. Secondly, when the users move, the target location moves along with them. This is accomplished by posting location updates to the server using the update request. Since this will happen asynchronously, it is possible that users at times will have different target locations locally. This error will however shrink over time when using the application, and since the location snaps to café locations the error will also round down.

### 2.1.2.4 Foursquare

To actually find cafés, the server uses the Foursquare API [6]. Everytime the geographical midpoint is calculated, the server does a lookup on the cafés in Foursquare’s database to find the café closest to the calculated location. The location of this café is what is actually sent to the clients, rather than the raw midpoint.

## 3 TESTING

For user testing, two test cases were tested. Firstly, trying to invite another person to use the app together with the tester, and being able to start navigation and, secondly, accepting an invitation from another user and starting the navigation.

### 3.1 Inviting

The testing of the inviting was mostly focused on the interaction with the app, how easy it was to use and to understand. The test participants were asked to start the app, use the app to invite another user and finally reacting on the push notification to start the actual navigation.

Participants were then asked if the application navigation was easy to follow, if the application when started had a clear call to action, and whether it was good or bad that the communication is using existing means of communication(i.e. text messaging and email), rather than implementing its own method.

### 3.2 Rsvp

As with the invitation testing, this test was mainly focused on evaluating the interaction in the application. Test participants were asked to act on a received email with the standard text and invite link. They were then asked whether the message was easy to understand, if they understood how to act on the message and if the implementation with links in messages rather than messages within the application worked well.

## 4 RESULTS

The invitation and RSVP tests were both performed with five different participants each.

### 4.1 Inviting

The participants unanimously found that the app had a clear call to action on the first screen. Most of them thought that the navigation was clear and easy to follow, but one screen in particular had some critique, namely the screen for selecting where to send the invitation message. One participant confused the list items with

text input fields and tried to input a phone number, since there wasn't a phone number present in the list.

Regarding the mode of communication, a few participants found that email is maybe not a good mode to use for this kind of more immediate messaging, as an email is more likely to be ignored than instant text messaging. One participant thought that the choice between email and text messaging is not really meaningful, and should maybe be a user setting.

One participant thought that sending a URL that you have not created yourself can feel like you are spamming the other user, especially if you are not used to iOS URL schemes.

## 4.2 Rsvp

Most participants liked the layout of the invitation message in general, one of the participants, however, wanted the link to accept to be display to be more prominent in order to provide a clearer call to action. Some also remarked that seeing the link without having use the app was confusing. One participant also thought that the email message was a little bit similar to spam, with just some text and a link, and that it should be clearer that the user is about to use the "Grab a Fika" application.

Some users also felt that the navigation screen was a little bit sparse, and that it's not really clear what the arrow is pointing to, especially if it's the first time you use the application. One user also thought that is hard to see if the other user is actually engaged, or if you are alone in using the application.

## 5 DISCUSSION

In general, the application was well received by testers, and the main part of the concept was implemented and working. There is still some edge cases to solve in regards to the application. One interesting result of the testing was that most testers found that it makes sense to actually use existing means of communication, rather than building a whole new communications system within the app, which was something the authors we're interested in understanding.

The testers did help with finding some issues with the application and the interface, such as the navigation screen not being clear to a novice user. One way to solve that would be to show an introduction the first time you run the app, but a better solution would be to make the application more informative, while still not give away too much information as this would work against the purpose of the app.

A clear difficulty and problem is that in using instant text messaging and email for inviting other users it is not very clear who the sender of the invite is, and what the purpose of the invite is. If a user that has never used the app before receives an invite and presses the link they will be taken to the app store, and especially to a user that is not well versed with iOS will have problems with keeping up with what is happening. It is the authors belief that this will work with a trust system, namely that if the user trusts whoever sent the invitation it is likely that they will actually respond to the invitation because of that trust.

It is however not clear from the testing whether this app would be something that users would like to use. The main point of the app is to be novel and make you find new places to go and new ways to get there, something that was hard to test in the limited scope of this project.

## 6 FUTURE WORK

As it stands the application only works between two people. This is to make the development easier and reduce edge cases eg. what would happen if one user accepts the invitation, and the other declines and so on. It also made the midpoint easier to calculate since the midpoint is the center of gravity of the shape created by the different positions, and with two positions that is a line.

Furthermore there is work to be done in addressing the issues and feedback collected during the user tests. These were minor but mainly concerned the navigation flow design within the application and also what information should and should not be on that navigation screen. It would also be good to conduct larger user tests with users actually navigating around a large area trying to find each other.

## 7 CONCLUSION

The application solves the problem of meeting up for a coffee with a friend in a novel way, if not effective. It can be perceived as a little bit contrived, but it's an interesting way to look at navigation. The application that was implemented worked well for the testing we did, testing that did provide useful feedback. It gave insight into things that were problematic with the interface, but also what was good. Using built in functionalities for messaging proved to be a good way of setting up a session, and helped in not forcing the user to create an account. But it also made it confusing for the receiving party if they were not introduced beforehand.

As a final remark, it is probably hard to beat a simple phone call agreeing on a time and place in efficiency, but sometimes one might want to try something new.

## 8 REFERENCES

1. Coe, Lewis. The telephone and its several inventors: A history. McFarland, 1995.
2. Katz, James E., and Mark Aakhus, eds. Perpetual contact: Mobile communication, private talk, public performance. Cambridge University Press, 2002.
3. Lundgren, Sus, and Olof Torgersson. "Bursting the Mobile Bubble." First International Workshop on Designing Mobile Face-to-Face Group Interactions, European Conference on Computer Supported Cooperative Work, ECSCW'2013. 2013.
4. Apple, Inc. "Advanced App Tricks". iOS Developer Library [https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/AdvancedAppTricks/AdvancedAppTricks.html#//apple\\_ref/doc/uid/TP40007072-CH7-SW50](https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/AdvancedAppTricks/AdvancedAppTricks.html#//apple_ref/doc/uid/TP40007072-CH7-SW50)
5. Apple, Inc. "iOS Human Interface Guidelines". iOS Developer Library. <https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/>
6. Foursquare, Inc. "Foursquare API". Foursquare for developers. <https://developer.foursquare.com/>