COLOCATION documentation

1 Introduction

This project revolves around the idea of implementing an accessible locationdatabase which can get userinput from mobile devices. We wanted to build a system that is flexible in such way that it can support multiple input sources which are not fysically connected to the system. The main user interface is a Flex-based websystem which will display your co-workers and buddy's last inputted location, on a digital map. By using the API of YahooMaps we are able to extend its broad functionallity. On the mobile end, we use J2ME Wireless Toolkit to develop a small mobile application in which users can input their location with a description of the current activity / interesting location. By detaching the user input from the system the users are more likely to share and log interesting locations because the mobile devices are becoming more and more capable in terms of connectivity and availibility.

2 Flex Implementation

Originaly we intended to use the OpenStreetMaps service for our mapdata, but since that service turned out to be quite bad, we switched to the Yahoo Maps service because it appeared to have a nice Flex API. However, during the programming process, this API didn't prove that nice at all, since it turned out to be a Flash API with a few ActionScript3 classes[1] to communicate with it. Because of the complexity of this communication kit, we had some problems implementing our application, but in the end we got it to work. The effect of this communication kit is that when the user opens our flex application, two Flash Virtual Machines get started, one for the Flash YahooMaps component, and one for our own application. The communication kit is then used to control the Flash component. From the perspective of our own application, the communication kit appears more or less as a collection of normal ActionScript 3 objects.

When a user opens our main application[2], the map is automatically centered around Amsterdam. The application then loads a list of all *locations* added by users, and each point is shown on the map as a Marker. When a user hovers his mouse over a marker, the addition date and time of the location is shown. Furthermore, when the user clicks a marker, a description is shown.

3 LOCATIONLET

Locationlet is a J2ME developed Midlet which is the main userinput program for our system. The program presents the user with a number of fields in which the location data can be inserted. The field asks the user for a userid which has to be registered at the serverside first. The second and third field are reserved for latitude and longitude coordinates. These should be updated when the action 'GetGPS' gets called, but due to implementation problems the gps functionallity is limited to the phones that workwith the amico-gps-sensor source. The current implementation After inputting the coordinates it is possible to supply a description of the current activity or interesting location. After input the user issues the 'Send' command to send the information to the php webscript which translates the input into a mysql data query, formatted to be read out by the Flex websystem.

While developing the midlet we stumbled across some difficulties that have proven themselves to take a lot of time. For starters the J2ME platform was new for all of us, so a lot of research has to be done to be able to perform simple tasks and it has been a while for us since we last programmed in java. Given the nature of our project, we tried to incorporate location data and webserver interaction on a mobile device which was quite a hassle to start programming with. The gps part of the location data was pretty hard as well, a lot harder then we expected. One of the reasons this didn't work out for us is because we are testing on a windows mobile 6.0 device with integrated qualcomm gps, which doesn't work with native gps api's. We have tried looking at te example code for the gps sensor but this wouldn't execute on the mobile phone we used, so maybe it could be possible for us to write a

[1] dubbed the Actionscript 3 Yahoo! Maps Communication Kit.

[2] to be found at http://peerdeman.homeip.net/imt/

program for other phones, but we wouldn't be able to test it ourselves.

4 Usage Scenario

1. Steve is starting his traintrip from Enkhuizen to Hoorn. While stepping into the train he takes out his mobile phone and logs his position with the Locationlet.
2. When Steve arrives at Hoorn he logs that he is taking the next train to Haarlem.
3. Danny, who is meeting Steve in the Haarlem Station, checks the Flex webfrontend and sees that Steve has started his trip on time, and goes off to the Haarlem Station.
4. Danny meets up with Steve and logs that they have met and take the train to Amsterdam together.
5. John checks the website and sees that his buddy's have taken their train and starts biking to the restaurant where they have will meet. John arrives and logs the location of the restaurant.

5 Future improvement

Registration online: It would be very functional and helpful for scalability to let users be able to register a userid in the system themselves, instead of hardcoding this in the mysql database. This would make for a good improvement of the website frontend, which could contain a more personalized version when a user is logged in after registering.

Addata online: As part of website frontend personalization and fast debugging it would be useful to create a webpages which facilitates the same functionality as the midlet, with the difference of running on an online webserver.

Photoupload: With the locationregistration in mind the adding of photoupload functionality would surely brighten up the websites content.

GetGPS: Ofcourse the getgps functionality is one of the biggest possible improvements. We have tried in a lot of different ways to get the gps working, but our experimenting mobile rig wouldn't pass its location data to the midlet. Further research and other test devices should make it possible to get this functionality working, but we decided to focus on making a working system which incorporates all the mentionned technologies.

6 Conclusion

As mentionned in the future improvements, the compatibility in our gps request position should be improved to support more devices such as our windows mobile integrated gps device. This project has certainly opened a lot of windows in our thinking process. The ease of implementing advanced functionality by integrating mobile applications with webservers is something we never would have thought of before this project. The flex environment gives its users a great variety of implementable technologies to quickly program robust and rich internet applications. Furthermore, the reading and the understanding of an existing websites API has proven itself very usefull, using already programmed libraries as part of your application is most certainly the future of programming applications.