

Mobile Services Project Report

Home Automation Mobile Application

[HAMA]

By: Colin Stanley and Eliazer Mbaeva



Faculty of Computer Science

Free University of Bolzano

7 September 2009

Table of Contents

<u>Topic</u>	<u>page</u>
The purpose:	3
Architecture:	3
Implementation:	4
The Code Structure	4
1. House MIDlet	4
2. Sender class	4
3. History class	4
The User Interfaces:	5
1. Launching HAMA	5
2. HAMA main form	5
3. The lights option	6
4. The curtains option:	8
5. The alarm option:	9

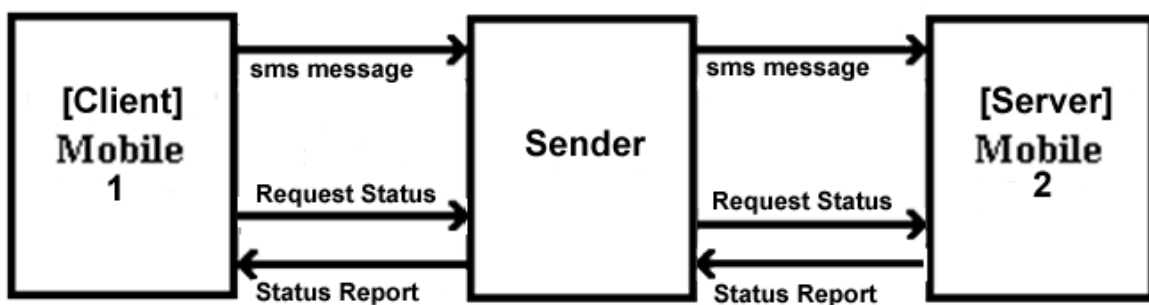
The purpose:

The purpose of this document is to outline and illustrate the development procedures carried out for **HAMA** , a mobile application designed to enable potential users who may be interested in controlling their home devices such as turning on the house lights, off/on setting of the alarm and closing curtains from their mobile phones.

Architecture:

HAMA is a client/server application that uses a simple architecture with two mobile phones, with mobile1 serving as a client and mobile2 serving as server. The communication between the two mobiles is carried out in a form of sms (the short service message). The event to send the message is triggered by the user with the client mobile and is transferred through the sender to the server.

Figure 1 Shows the HAMA system architecture, gives a picture of how the messages are to be send from the client to the server



Implementation:

The Code Structure

The application has a **home package** which consists of two classes:

1. House MIDlet

- Implementing the graphic user interfaces of the application, has a main form with a choicegroup mechanism that's a host of the house items [Lights, Curtain and Alarm]
- The house items provide users with selectable options available for possible adjustments, users can select between light 1 through light 4 which they switch on/off. A number of curtains are available and they could be adjusted by a gauge, the alarm has an On/Off setter.

2. Sender class

- This is the application's message sending component. One of the methods of this class, the sendMsg takes three parameters (Receiver, Port and Message), its purpose is to initialize the message receiver, the port, the message to be sent. It is in this class where the thread is initiated.
- Also contained in the sender class is the sendSMS, it is responsible for sending the sms. It defines the address to which the sms is to be sent and the connection.

The **model package**:

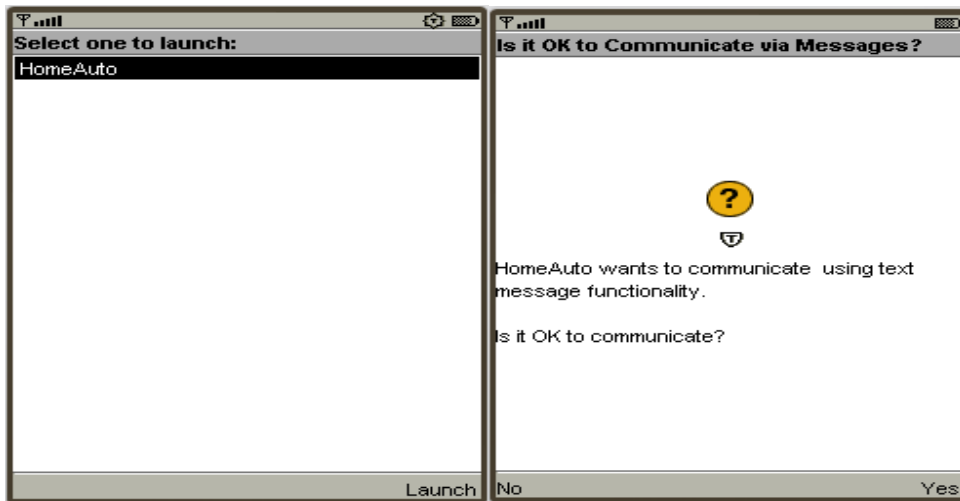
3. History class

The model package implements the **History class** which is a host of a RecordStore mechanism responsible for record storage management, capable for loading, saving and retrieving records.

The User Interfaces:

1. Launching HAMA

Figure 2 HAMA after ran



When the application is launched, the application's main form with the capability to allow user to select the house items is prompted.

2. HAMA main form

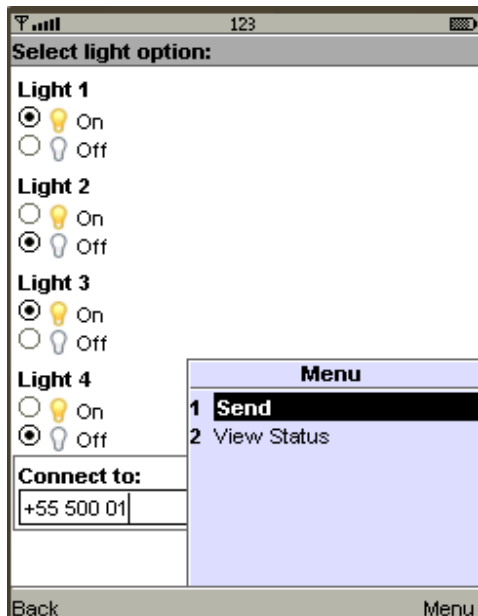
Figure 3 The main form



The client contains a main form that has a list of three house items; these are lights, curtains and alarm.

3. The lights option

Figure 4 The lights form



When selected from the main menu, the light option displays a list of four light options (Light 1 through light 4), each of which has an On/Off setter.

By default, all the lights are off and their state change if the On button is selected.

Warning message:

If the user did not input the mobile number to connect the message is not send and a warning alert is displayed.

Figure 5 House lights status before sending a sms

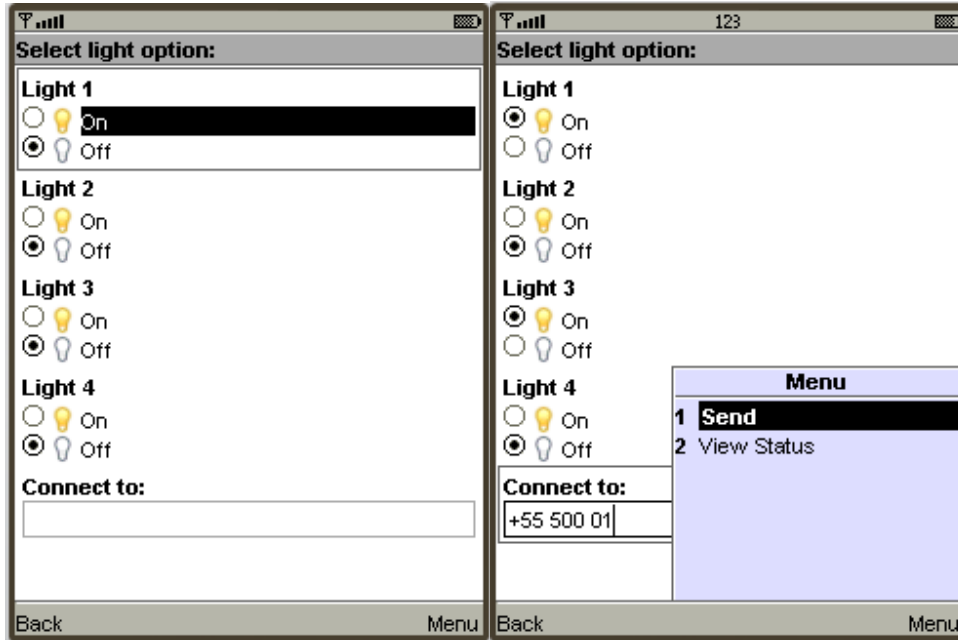


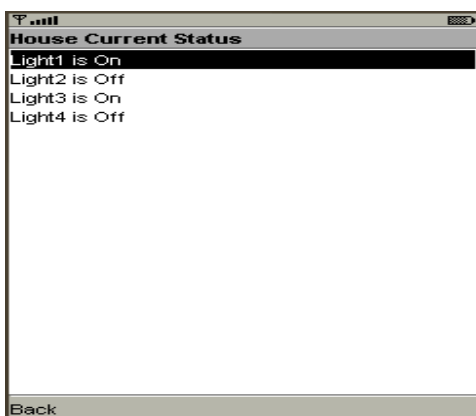
Fig 1.3 Default status of house lights viewed from the server screen on the left, the screen on the right shows a client mobile screen(+555 0000) ready for sending a sms to a mobile phone whose number is +555 0001

When the message is sent from mobile +555 0000, the status of the screen will be updated and the selected status from the sender should be reflected on the server.

Viewing the lights status:

Generally, features statuses could be viewed by hitting the view status command on the menu.

Figure 6 light status results screen



4. The curtains option:

With four curtains (First, Second, Third and Fourth) available for controlling, the curtains statuses are set to 0 by default and so the gauge bars symbolizing their status is all clear at the beginning. At that state, the curtains are fully open. The curtain is fully closed if the gauge bars are all shaded/filled, otherwise the curtain is as wide/narrow open as the gauge shows it.

Figure 7 curtains screen at start

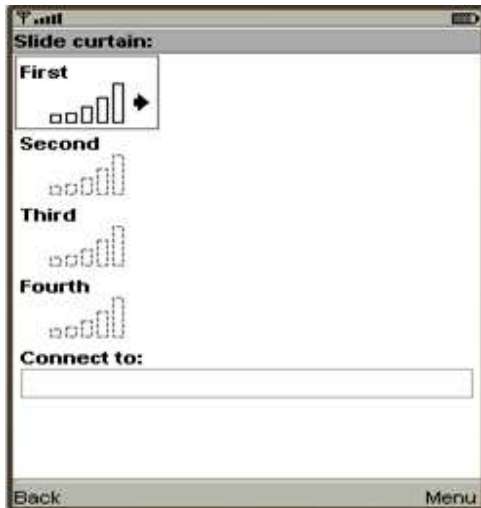
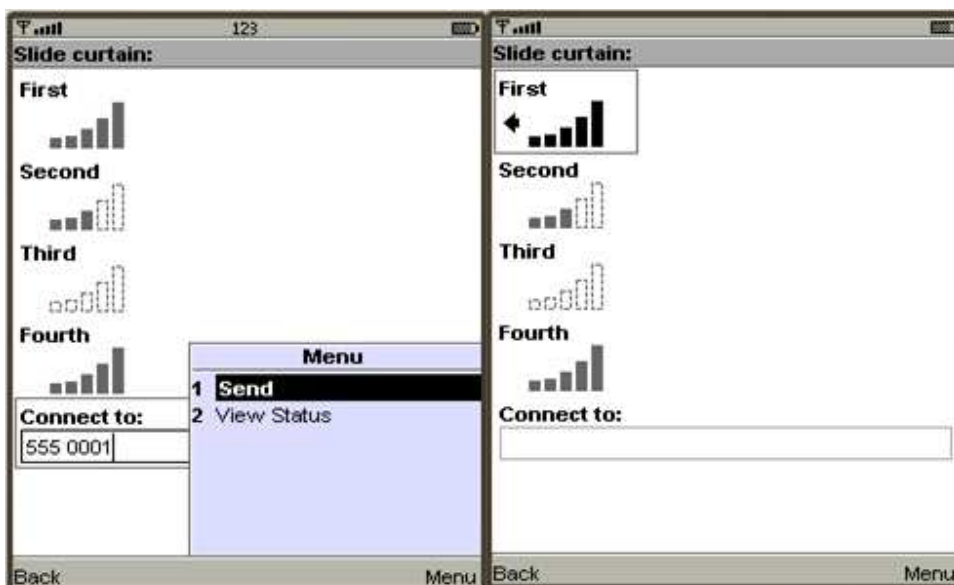
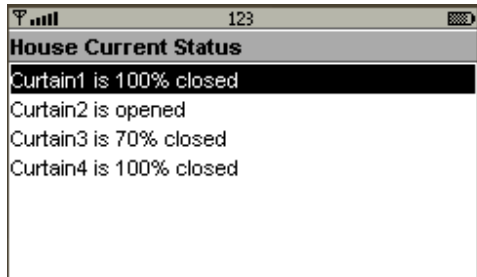


Figure 8 shows a left screen set for sending a message from the client to the server, the right screen is an updated version of the server after the message has been received



The status screen for curtains:

Figure 9 shows an example of the curtain status



5. The alarm option:

The alarm option activity fashion is the same as the light hence its functionality is also based on setting it to an Off/On status. The status is off by default. The alarm can either be set for the windows or for the door or for both.

Figure 10 shows the alarm screen at start

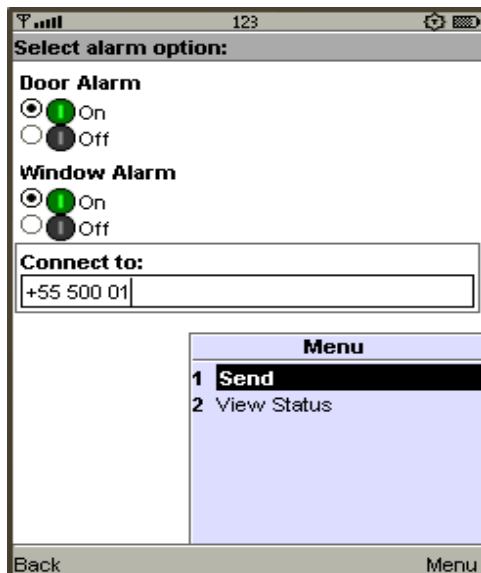


Figure 11 shows a sender screen on the left and the updated screen of the alarm status

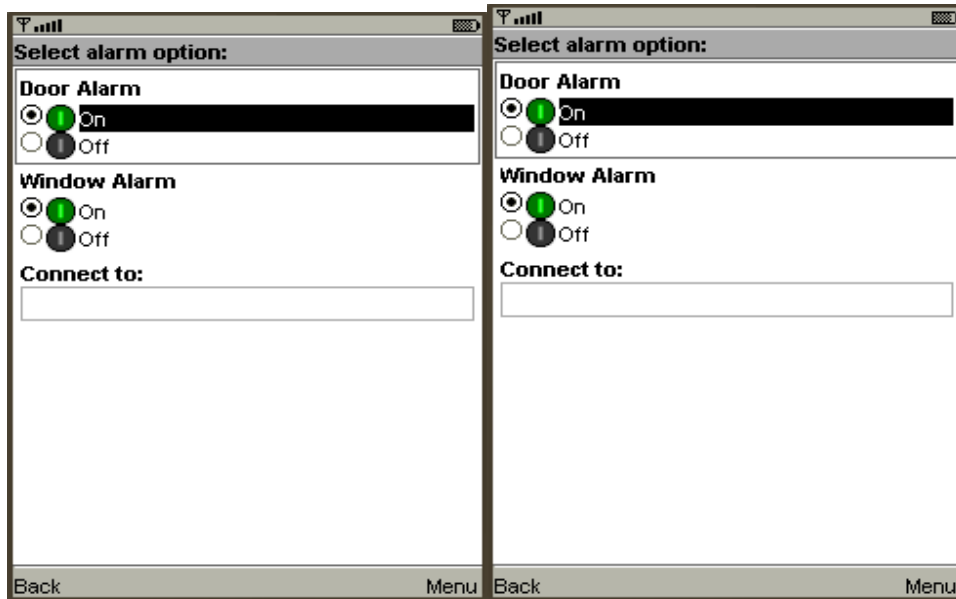


Figure 12 Alert for alarm request

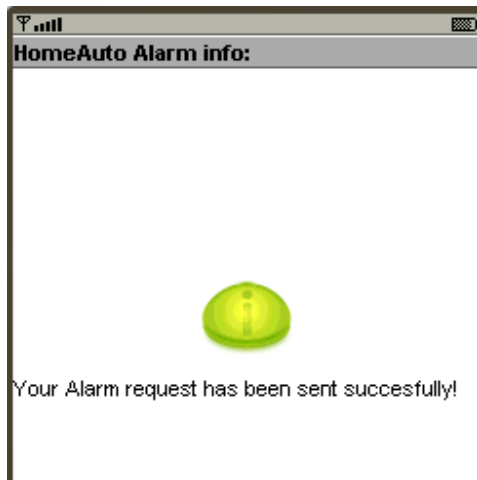


Figure 13 shows the viewed status of the house alarm devices



References:

Mobile Services Slides, Professor: [Francesco Ricci](#)

<http://www.inf.unibz.it/~ricci/MS/index.html> last accessed: 7 September 2009

Sing Li and Jonathan Knudsen, Beginning J2ME, (3rd Ed.) Apress, 2005