COSC2543 MOBILE APPLICATION PROGRAMMING

TUTORIAL WEEK 2: SIMPLE MIDLET WITH UI SOLUTION

MainScreen

```
package thegame;

import javax.microedition.lcdui.*;
import javax.microedition.lcdui.Alert;
import javax.microedition.midlet.*;
import javax.microedition.rms.RecordStoreException;

public class MainScreen extends MIDlet implements CommandListener {

    private Display display;
    private List startupList;
    private Command exit;
    private String[] ELEMENTS = {"New", "Highscore", "Settings"};

    public MainScreen() {
        startupList = new List("The Game", List.IMPLICIT, ELEMENTS, null);
        exit = new Command("Exit", Command.EXIT, 0);
        startupList.addCommand(exit);
        startupList.setCommandListener(this);
    }

    public void startApp() {
        if (display == null) {
            display = Display.getDisplay(this);
        }

        display.setCurrent(startupList);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
        // Close connectionto recordstore
    }

    public void commandAction(Command c, Displayable d) {
        if (c == exit && d == startupList) {
            destroyApp(false);
            notifyDestroyed();
        } else if (c == List.SELECT_COMMAND && d == startupList) {

            String chosen = startupList.getString(startupList.getSelectedIndex());
            if(chosen.equals("Settings")) {
                display.setCurrent(new SettingsScreen(this));
            } else {
                Alert alert = new Alert("Title", chosen, null, AlertType.INFO);
                display.setCurrent(alert);
            }
        }
    }

    public void showMainScreen() {
        display.setCurrent(startupList);
    }
}
```

SettingsScreen

```
package thegame;

import javax.microedition.lcdui.*;
public class SettingsScreen extends Form implements CommandListener {

    private final String TITLE = "Settings";
    private final String SAVETOSERVER = "Store name and location automatically to server";
    private final String NAME = "Your Name: ";
    private final String BACK = "Back";
    private final String HIGHSCORE = "Highscore:";

    private TextField name;
    private ChoiceGroup saveToServer;
    private StringItem highScore;
    private Command back;
    private MainScreen host;
    public SettingsScreen(MainScreen host) {
        super("");
        setTitle(TITLE);
        this.host = host;

        String[] items = {SAVETOSERVER};

        name = new TextField(NAME, "Untitled", 50, TextField.ANY);
        saveToServer = new ChoiceGroup("", Choice.MULTIPLE, items, null);
        highScore = new StringItem(HIGHSCORE, "0");
        back = new Command(BACK, Command.BACK, 0);

        append(name);
        append(saveToServer);
        append(highScore);

        addCommand(back);
        setCommandListener(this);

    }

    public void commandAction(Command c, Displayable d) {
        if (c == back && d == this) {
            host.showMainScreen();
        }
    }
}
```