

```
package fi.paatti.mobile.views.progressview;

import com.vaadin.data.Item;
import com.vaadin.data.util.sqlcontainer.SQLContainer;
import com.vaadin.ui.Label;
import fi.paatti.mobile.applicationinfo.CssStyleNames;
import fi.paatti.mobile.applicationinfo.Resources;
import fi.paatti.mobile.mobileview.MobileViewLogic;
import fi.paatti.mobile.views.mobilecontentview.MobileContentView;
import fi.paatti.mobile.paattiapplication.MobileViewHandler;
import fi.paatti.mobile.views.eventlistview.EventListItem;
import fi.paatti.mobile.views.eventlistview.EventListView;
import fi.paatti.mobile.views.listview.ListItem;
import fi.paatti.mobile.views.listview.ListView;
import fi.paatti.paattidatabaseutils.names.PaattiColumnNames;
import fi.paatti.paattidatabaseutils.names.PaattiTableNames;
import java.util.HashSet;

/**
 * A view that shows the progress of the user.
 * <p/>
 * For now all the events that user has done, will be shown here.
 *
 * TODO: The whole logic in this view should be re-designed. Current
 * implementation is just for showing the events that the user has done.
 * <p/>
 * @author Lauri Satokangas, lauri.n.satokangas@student.jyu.fi
 */
public class ProgressView extends ListView {

    private static final long serialVersionUID = 1L;

    /**
     * Constructor.
     * <p/>
     * @param viewHandler The applications view handler.
     */
    public ProgressView(MobileViewHandler viewHandler) {
        super(viewHandler);
        createUi();
    }

    private void createUi() {
        updateFooterAndHeader();
    }
}
```

```
}
@Override
public void setContentViewModel(MobileViewLogic logic) {// Not in use
}
/**
 * Gets the done events from database and creates the list items from those.
 */
private void createListItemsFromDb() {
    // Remove old components before adding new.
    this.removeAllComponents();
    SQLContainer listItemsContainer;

    // This is just a temporary solution.
    // All the taskdata will be get from the database, but only the event titles will be shown.
    listItemsContainer = getViewHandler().getDbService().getUserEventData(getViewHandler().getUserID());

    if (listItemsContainer.size() > 0) {
        Object itemId = listItemsContainer.firstItemId();
        Item eventItem;

        // Use a HashSet to keep record of all the execution times of the events.
        // It is HIGHLY unlikely that a user has finished two different events at the same time.
        HashSet<String> eventTimesMap = new HashSet<String>();

        while (itemId != null) {
            eventItem = listItemsContainer.getItem(itemId);
            String eventTime = eventItem.getItemProperty(PaattiColumnNames.TASKDATA_executionTime).getValue().toString();

            if (!eventTimesMap.contains(eventTime)) {
                ListItem listItem = new ListItem(eventItem, PaattiColumnNames.EVENT_eventID, PaattiColumnNames.EVENT_title,
                    PaattiColumnNames.TASKDATA_executionTime);

                listItem.setOpenButtonEnabled(false);
                listItem.setStyleName(CssStyleNames.PROGRESSVIEW_LISTITEM);

                this.addComponent(listItem);
                eventTimesMap.add(eventTime);
            }

            itemId = listItemsContainer.nextItemId(itemId);
        }
    }
    // TODO: this is also just a temporary solution.
}
```

```
Label noEventData = new Label("sinulla ei ole suoritettuja tapahtumia.");

// TODO: fix proper styleName
noEventData.setStyleName(CssStyleNames.EVENTLISTVIEW_HEADER_LABEL);
this.addComponent(noEventData);
}
}

@Override
public void updateFooterAndHeader() {
    getViewHandler().setHeaderText("Suoritetut tapahtumat");
    getViewHandler().setFooterButtonIcon(FOOTER_BUTTON_LEFT, Resources.getThemeResource(Resources.ICON_ARROW_LEFT_64));
    getViewHandler().setFooterButtonEnabled(FOOTER_BUTTON_MIDDLE, false);
    getViewHandler().setFooterButtonEnabled(FOOTER_BUTTON_RIGHT, false);

    createListItemsFromDb(); // not really a part of footer or header..
}
}
```