

# Project Ivanhoe

## Design Document

Group 16

Rebecca Everett

Tobias Hassellöf

Henrik Törnvall

Johan Renner

Martin Waara-Grape

## 5.5 Detailed Design

| AnalyseData   |
|---|
| +compareWithBudget(Date timeperiod): ArrayList<br>+produceDiagramData(ArrayList list1, ArrayList list2, ArrayList list3): ArrayList |

### Method: compareWithBudget()

#### *Parameters*

Date timeperiod

#### *Return value*

ArrayList

#### *Description*

Computes and structures data for the table in the “Visningsläge” tab in the GUI.

#### *Called by*

Profile.processShowData()

#### *Calls*

Profile.getBudgetCollection()

Profile.getDatedEventCollection()

*Reference to RD:* 6.1.1.6.1, 6.1.1.6.2

### Method: produceDiagramData()

#### *Parameters*

ArrayList list1

ArrayList list2

ArrayList list2

#### *Return value*

ArrayList

#### *Description*

Computes and structures data for the table or diagram in the “Diagram” tab in the GUI.

#### *Called by*

Profile.processDiagramData()

#### *Calls*

Parameters.getParameters()

Profile.getBudgetCollection()

Profile.getDatedEventCollection()

*Reference to RD:* 6.1.1.7

## Profile

-budgetCollection: <ArrayList>Budget  
-datedEventCollection: <ArrayList>DatedEvent  
-name: String  
-file: File  
-changedSinceLastSave: Boolean

+Profile(String name)  
+getBudgetCollection(): <ArrayList>Budget  
+getDatedEventCollection(): <ArrayList>DatedEvent  
+getFile(): File  
+setFile()  
+getChangedSinceLastSave(): Boolean  
+setChangedSinceLastSave(Boolean arg)  
+processShowData(Date timeperiod): ArrayList  
+processDiagramData(ArrayList list1, ArrayList list2, ArrayList list3): ArrayList

### Method: getBudgetCollection()

#### *Parameters*

None

#### *Return value*

<ArrayList>Budget

#### *Description*

Returns the budget data for the active budget.

#### *Called by*

AnalyseData.compareWithBudget()  
AnalyseData.produceDiagramData()

#### *Calls*

None

*Reference to RD:* -

### Method: getDatedEventCollection()

#### *Parameters*

None

#### *Return value*

<ArrayList>DatedEvent

#### *Description*

Returns the DatedEvent data for the active profile.

*Called by*

AnalyseData.compareWithBudget()  
AnalyseData.produceDiagramData()

*Calls*

None

*Reference to RD: -*

**Method: getFile()**

*Parameters*

None

*Return value*

File

*Description*

Get the File this Profile was most recently read from or saved to.

*Called by*

FileSystem.saveFile()

*Calls*

None

**Method: setFile()**

*Parameters*

None

*Return value*

void

*Description*

Set the File this Profile was most recently read from or saved to.

*Called by*

FileSystem.readFile()

*Calls*

None

*Reference to RD: -*

### **Method: getChangedSinceLastSave()**

*Parameters*

None

*Return value*

boolean

*Description*

Find out whether this profile has been changed since last open / save.

*Called by*

Controller.doSave()

*Calls*

None

*Reference to RD:* -

### **Method: setChangedSinceLastSave()**

*Parameters*

Boolean arg

*Return value*

void

*Description*

Record a change in the saved status of this profile.

*Called by*

Controller.doSave()

*Calls*

None

*Reference to RD:* -

### **Method: processShowData()**

*Parameters*

Date timeperiod

*Return value*

ArrayList

*Description*

Requests class AnalyseData to process data to be shown in the “Visningsläge” tab in the GUI.

*Called by*

Controller.doAnalyseData()

*Calls*

None

*Reference to RD:* 6.1.1.6.1

### **Method: processDiagramData()**

*Parameters*

ArrayList list1

ArrayList list2

ArrayList list3

*Return value*

ArrayList

*Description*

Requests class AnalyseData to process data to be shown in the “Diagram” tab in the GUI.

*Called by*

SystemController.doAnalyseData()

*Calls*

None

*Reference to RD:* 6.1.1.7

| <b>EconomicEvent</b>   |
|--|
| <b>-type:string</b><br>-category:String<br>-name:String<br>-value:int  |
| <u>+EconomicEvent(type, category, name, value)</u><br>+getType():String<br>+getCategory():String<br>+getName():String<br>+getValue():int<br>+getEconomicEvent():EconomicEvent<br>+setType(String):<br>+setCategory(String):<br>+setName(String): |

+setValue(int):

**Method: getType()**

*Parameters*

None

*Return value*

String

*Description*

Returns the Type for the EconomicEvent

*Called by*

Budget.addEconomicEvent()

*Calls*

None

*Reference to RD: -*

**Method: getCategory()**

*Parameters*

None

*Return value*

String

*Description*

Returns the Category for the EconomicEvent

*Called by*

Budget.addEconomicEvent()

*Calls*

None

*Reference to RD: -*

**Method: getName()**

*Parameters*

None

*Return value*

String

*Description*

Returns the Name of the EconomicEvent

*Called by*

Budget.addEconomicEven()

*Calls*

None

*Reference to RD: -*

### **Method: getValue**

*Parameters*

None

*Return value*

int

*Description*

Returns the Type for the EconomicEvent

*Called by*

Budget.addEconomicEvent

*Calls*

None

*Reference to RD: -*

### **Method: getEconomicEvent**

*Parameters*

None

*Return value*

EconomicEvent

*Description*

Returns an EconomicEvent

*Called by*



Budget.addEconomicEvent()  
Budget.updateEconomicEvent()

*Calls*  
None

*Reference to RD:* -

### **Method: setType**

*Parameters*  
String

*Return value*  
None

*Description*  
Sets the Type for an EconomicEvent.

*Called by*  
Budget.addEconomicEvent()  
Budget.updateEconomicEvent()

*Calls*  
None

*Reference to RD:* 6.1.1.2.1.2

### **Method: setCategory**

*Parameters*  
String

*Return value*  
None

*Description*  
Sets the Category for an EconomicEvent.

*Called by*  
Budget.addEconomicEvent()  
Budget.updateEconomicEvent()

*Calls*  
None

*Reference to RD:* 6.1.1.2.1.1

**Method: setName***Parameters*

String

*Return value*

None

*Description*

Sets the Name for an EconomicEvent.

*Called by*

Budget.addEconomicEvent()

Budget.updateEconomicEvent()

*Calls*

None

*Reference to RD:* 6.1.1.2.1.1

**Method: setValue***Parameters*

Int

*Return value*

None

*Description*

Sets the value of an EconomicEvent.

*Called by*

Budget.addEconomicEvent()

Budget.updateEconomicEvent()

*Calls*

None

*Reference to RD:* 6.1.1.2.1.1

|                         |
|-------------------------|
| <b>SystemController</b> |
|-------------------------|

## -ivanhoeGUI: IvanhoeGUI

-fileSystem: FileSystem

-profile: Profile

+SystemController(FileSystem filesystem):

+doSave(gui)

+doCreateProfile(gui)

+doDeletProfile(gui)

+doCreateBudget(gui)

+doEditBudget(gui)

+doAddEconomicEvent(gui)

+doAddDatedEvent(gui)

+doDeleteEvent(gui)

+doEditEconomicEvent(gui)

+doEditDatedEvent(gui)

+doAnalyseData(gui)

+doOpenProfile

### The following applies to all “do” methods:

#### *Parameters*

IvanhoeGUI gui

#### *Return value*

Void

#### *Description*

Requests different actions from the classes that Controller controls.

#### *Called by*

GUI

#### *Calls*

Various methods in the appropriate classes.

*Reference to RD:* -

## **Budget**

-budget: ArrayList <EconomicEvent>

-name: String

-timespan:int

```
+Budget(name, timespan)
+addEconomicEvent(String type, String category, String name, int value)
+updateEconomicEvent(type, category, name, value)
+removeEconomicEvent()
+getBudget(): Budget
+sortByCateogry()
+sortBySum()
+sortByName()
+sortByType()
+sortByValue()
```

### **Method: addEconomicEvent**

#### *Parameters*

String Type  
String Category  
String Name  
int value

#### *Return value*

Void

#### *Description*

Adds an EconomicEvent to the budget.

#### *Called by*

SystemController.doAddEconomicEvent()

#### *Calls*

EconomicEvent()

*Reference to RD:* 6.1.1.2.1

### **Method: updateEconomicEvent**

#### *Parameters*

type, category, name, value

#### *Return value*

Void

#### *Description*

Updates an EconomicEvent in the budget.

#### *Called by*

SystemController

#### *Calls*

EconomicEvent

*Reference to RD:* 6.1.1.3

**Method: removeEconomicEvent**

*Parameters*

None

*Return value*

Void

*Description*

Deletes an EconomicEvent from the budget.

*Called by*

SystemController

*Calls*

EconomicEvent

*Reference to RD:* 6.1.1.3

**Method: getBudget**

*Parameters*

None

*Return value*

Void

*Description*

Returns a budget to the caller.

*Called by*

Profile.getBudgetCollection()

*Calls*

None

*Reference to RD:* -

**Method: sortByCategory**

*Parameters*

None

*Return value*

Void

*Description*

Sorts the EconomicEvents in the budget after Category.

*Called by*

SystemController

*Calls*

EconomicEvent.getCategory()

*Reference to RD:* -

**Method: sortBySum**

*Parameters*

None

*Return value*

Void

*Description*

Sorts the EconomicEvents in the budget after the sum.

*Called by*

SystemController

*Calls*

EconomicEvent.getValue()

*Reference to RD:* -

**Method: sortByName**

*Parameters*

None

*Return value*

Void

*Description*

Sorts the EconomicEvents in the budget after the name.

*Called by*  
SystemController

*Calls*  
EconomicEvent.getName()

*Reference to RD:* -

### **Method: sortByType**

*Parameters*  
None

*Return value*  
Void

*Description*  
Sorts the EconomicEvents in the budget by Type.

*Called by*  
SystemController

*Calls*  
EconomicEvent.getType()

*Reference to RD:* -

### **Method: sortByValue**

*Parameters*  
None

*Return value*  
Void

*Description*  
Sorts the EconomicEvents in the budget after Value.

*Called by*  
SystemController

*Calls*  
EconomicEvent.getValue()

*Reference to RD: -*

| <b>FileSystem</b>   |
|---|
| +FileSystem()<br>+readFile(File file): Profile<br>+saveFile(File file, Profile profile) |

**Method: readFile()**

*Parameters*

File file

*Return value*

Profile

*Description*

Read a stored file.

*Called by*

SystemController.doOpenProfile()

*Calls*

None

*Reference to RD: -*

**Method: saveFile()**

*Parameters*

File file

Profile profile

*Return value*

Void

*Description*

Save a profile to a file.

*Called by*

SystemController.doSave()

*Calls*

None

*Reference to RD: -*



|                               |
|-------------------------------|
| <b>IvanhoeApplication</b>     |
| -fileSystem: FileSystem       |
| -controller: SystemController |
| +main()                       |
| +quitApplication()            |

**Method: main()**

*Parameters*

None

*Return value*

Void

*Description*

Initiates the application.

*Called by*

None

*Calls*

None

*Reference to RD: -*

**Method: quitApplication()**

*Parameters*

None

*Return value*

Void

*Description*

Quits the application.

*Called by*

SystemController.doQuit()

*Calls*

None

*Reference to RD: -*

|   |
|---|
| <b>IvanhoeGUI</b>   |
| -controller: SystemController<br>-profile: Profile<br>-showTable: JTable  |
| +IvanhoeGUI(SystemController controller, Profile profile):<br>+getProfile(): Profile<br>+setProfile(Profile profile)<br>+reportError(String message)<br>+update(Observable o, Object obj) |

### **Method: getProfile()**

#### *Parameters*

None

#### *Return value*

Profile

#### *Description*

Accessor for the profile this GUI displays.

#### *Called by*

IvanhoeGUI()

#### *Calls*

Profile.getProfile()

*Reference to RD:* -

### **Method: setProfile()**

#### *Parameters*

Profile profile

#### *Return value*

Void

#### *Description*

Changes the profile this GUI displays.

#### *Called by*

IvanhoeGUI()

#### *Calls*

None

*Reference to RD: -*

**Method: reportError()**

*Parameters*

String message

*Return value*

Void

*Description*

Reports an error represented by a string.

*Called by*

IvanhoeGUI

*Calls*

None

*Reference to RD: -*

**Method: update()**

*Parameters*

Observable o

Object obj

*Return value*

Void

*Description*

Method required by the Observer interface - update the display in response to any change in the profile

*Called by*

IvanhoeGUI

*Calls*

None

*Reference to RD: -*

|                   |
|-------------------|
| <b>DatedEvent</b> |
|-------------------|

-date: Date

Class also includes the variables methods as EconomicEvent.

+DatedEvent(Date date)

+getDate(): Date

Class also includes the same methods as EconomicEvent.

### **Method: getDate()**

#### *Parameters*

None

#### *Return value*

Date date

#### *Description*

Returns the date value.

#### *Called by*

Profile.getDatedEventCollection()

#### *Calls*

None

*Reference to RD:* 6.1.1.2.1.1

## 5.6 Package Diagram

There are two levels in the package diagram. The top level consists of:

- The IvanhoeApplication, which instantiates the SystemController and the IvanhoeGUI,
- The FileSystem that controls whether the Model is saved since the last change
- The SystemController, which performs the events that occur in the IvanhoeGUI, creates and manages the Profile, uses the FileSystem to read and write files and is able to quit the application by sending a command to the IvanhoeApplication.
- The IvanhoeGUI that provides the user with a graphical interface and is an observer of the Profile.

The second level is the Model package that consists of:

- The Profile, which uses the inner class AnalyseData to analyse the data and creates and keeps track of DatedEvents and Budgets
- The Budget that creates and keeps track of EconomicEvents.
- The DatedEvent and the EconomicEvent keeps track of data concerning the income or expense.
- The AnalyseData that calculates the difference between EconomicEvents and DatedEvents

