Project Ivanhoe

Group 16
Rebecca Everett
Tobias Hassellöf
Henrik Törnvall
Johan Renner
Martin Waara-Grape

User requirements – functional requirements:

1. The system shall support several users by allowing the user to create profiles.
The profile is used to link the budgets created to a specific user since the system will be able to have several users.

2. The user shall be able to create, manage and save several budgets.
The budget consists of expected incomes and expenditures that the user specifies when creating it.

3. The user shall be able to categorise budgeted and actual cash flow.
The categories will help the user monitor specific expenses such as food expenses or car expenses. This will provide the user with a clear understanding of where the money goes.

4. The user shall be able to record actual income and expenses.
When the user has inputted the cash flow, he/she can then compare this with the budget for a chosen period.

5. The user shall be able to compare income and expenses with the budget.
The user will be provided with a textual and a graphical presentation of how the actual cash flow coincides with the budget.

6. The user shall be able to write a to-do list.
The to-do list should consist of the due dates of the upcoming expenses, which the user can use when paying the bills.

7. The system shall provide the user with a textual overview of budgeted and actual cash flow.
The textual overview will provide the user with information such as in what areas he/she has been overspending.

8. The user shall be able to compare different time-periods.
In order to better understand how for example the user's expenses vary, the user should be able to monitor all expenses or a category of expenses over a chosen period of time.

9. The system shall provide the user with information about due-dates when the program starts.
The due dates will inform the user of the expenses that should be paid within a short period of time.

10. The user shall be able to assign a due date for each item.
The due date is used to inform the user of upcoming payment due dates as well as design the economic event to the right period.

11. The system shall provide the user with a graphical overview of budgeted and actual cash flow.
The graphical overview will provide the user with an overview and information such as in what areas he/she has been overspending.

12. The system shall provide the user with a help file.
The help file will work as a source of information when learning the program and needing help with common problems.

13. The user should be able to compare different expense items for different time-periods.
The comparison will give the user an overview of how his/her expenses vary over time and of the progress he/she is making.

System requirements – functional requirements:

1.1 The system shall take an input from the user when requesting his/her name for the profile.
1.2 The system shall link each budget to the active profile by assigning it an attribute.
1.3 The system shall link each to-do list to the active profile by assigning it an attribute.
1.4 The system should allow the user to change to another profile by choosing this in a menu bar. The user should then be prompted to save all changes and existing work.
1.5 The system shall provide the user with a list of existing profiles.

2.1 The system shall take an input from the user when requesting the name of the budget.
2.2 The system shall take inputs from the user regarding expected incomes and expenses when creating a budget.
2.3 The system shall take inputs from the user regarding dates of the cash flow.
2.4 The system shall provide the user with a list of predefined categories for each cash flow.
2.5 The system shall provide the user with an opportunity to define new categories when creating a budget or recording cash flow.
2.6 The system shall be able to import saved budgets.

3.1 The system shall be able to generate a budget from the information provided by the user.
3.2 The system shall provide the user with a list of predefined categories when creating a budget.
3.3 The system shall provide the user with an opportunity to define new categories when the user creates a budget as well as records cash flow.
3.4 The system shall take inputs from the user when the user defines a new category.
3.5 As soon as new category is defined, the system should add the category to the list of categories without a reboot of the system.

4.1 There shall be inputs fields in which the user can type in the name, type and amount of actual income and expenses.
4.1.1 This data shall be saved.

5.1 The system shall compute the difference between the budget data and actual outcome data of a specified time period.
5.1.1 The system shall display this information to the user.
5.1.2 The system shall highlight over-expenditures (negative amounts).

6.1 The system should create a to-do list from the upcoming due dates specified by the user.
6.2 The system should be able to print the to-do list on an installed printer.
6.3 The system should save the to-do list as a text file.
6.4 The system should allow the user to add and remove items from the to-do list.

7.1 The system should generate a table containing income and expense items using the stored budget data.
7.2 The system should generate a table containing income and expense items using the stored outcome data.
7.3 The system should calculate the difference between the sums in the tables described in 7.1 and 7.2.

8.1 The system should be able to show two different budgets side by side.
8.2 The system should provide a function to browse budgets sequentially.
8.3 It should be possible to overlay two or more graphs in the same diagram, using different colours.

9.1 The system shall scan the current budget and search for unpaid bills.
9.2 The system shall provide an alert box displaying these items.
9.3 There should be input fields to mark the items as paid.

10.1 The system shall take an input from the user for each expenditure and income inputted.
10.2 The system shall by default have the date of the current input as the due date.
10.3 The system shall allow the user to change the due date after the initial input.

11.1 The system shall display both budgeted and actual cash flow in the graphical user interface.
11.2 The system shall group and display the different inputted items according to their stated categories.
11.3 The system shall provide a summary of the budgeted and actual cash flow.
11.4 The system shall be able to display different financial items in different time spans.
11.5 The system shall highlight the categories where the financial outcome diverges from budgeted value.

12.1 The system should provide the user with menu option to view the help file.
12.2 The help file should be indexed on both financial terms and practical steps of how to achieve different tasks in the program (i.e. creating a budget).
12.3 The help file will be formatted as a HTML-file.

13.1 The system shall take as input an expense item and a range of dates.
13.2 The system shall compare the expenditures on the expense item for the different time periods.
13.3 The system shall display the new data as text.
13.4 The system should be able to display the data as graphical information.

Non-functional requirement

1. The programming language shall be Java.
2. The program shall be easy to learn and have an average learning curve of 20 minutes.
3. The program's calculations shall be 100% correct.
4. The program shall start within 10 seconds.
5. The program shall run on Windows XP using Java Runtime Environment JRE 1.5
6. The program shall be delivered on date (unknown at the time of writing)

Use Case UC1 – Record expense

Primary Actor:
The student
Stakeholders and Interests:
- User: Wants to be able to record an expense in his/her profile. The process should be intuitive and there should be a possibility to cancel. The user wants to be able to add a note to the expense and categorise it.
Preconditions:
The computer is turned on and the student has started the program. The user already has a profile.
Success Guarantee (postconditions): The expense is registered on the profile and can be compared against a budget or visualised on a timeline.
Main Success Scenario (or Basic Flow):

1. The student chooses his/her profile in a menu.
2. The student chooses to record the expense in a menu
3. A new dialog is shown by the system.
4. The student inputs the amount of the expense.
5. The student chooses a category for the expense.
6. The student inputs a short note explaining the expense.

The system shows the current date as the default value on the date of the expense.

7. The student changes the date to the previous month.

The student repeats 4-7 until all the expenses has been inputted.

8. The student chooses "Done" in the dialog.
9. The dialog is closed and the expenses are automatically saved by the system to the current profile.
10. The student exits the program.

Extensions (or Alternative Flows):
*a. At any time, System fails:
If the system fails and the program shuts down then only the saved data will be available. If the student hasn't chosen "Done" in the dialog, then the information is lost.
4a. The student inputs a negative amount of the expense.

1. The system rejects the amount and registers the absolute value.

5a. The category that the user wants to use does not exist.

1. The user adds another category.
2. The user chooses this category for the expense.

5b. The student doesn't know what category to use.

1. The student chooses miscellaneous, which is a predefined category.
2. The student decides to change the category at a later time.

6a. The student chooses not to enter a note for the expense.
7a. The student doesn't change the default date since the expense occurred today.
8a. The student doesn't chose "Done" and chooses "Cancel" instead.

1. The dialog is closed and the recorded data is discarded.
2. The student exits the program.

8b. The student wants to record more than 10 expenses and has already inputted 10.

1. The student chooses save.
2. The system saves the expenses to the current profile.
3. The information in the dialog is cleared and the system is ready to receive new inputs.
4. The student repeats 4-7.

9a. The student hasn’t inputted an amount and /or chosen a category for one or more expenses.

5. The dialog is not closed.
6. The system informs the student that there is information missing.
    a. The student chooses cancel.
        i. The dialog is closed and the information is discarded.
    b. The student inputs the missing information.
        i. The student chooses “Done” again.
7. The dialog is closed and the expenses are automatically saved by the system to the current profile.

Use Case UC2 – Create profile

Stakeholder & primary actor: Mother of family: wants to create a profile.
Preconditions: Computer is on and program is started.
Success Guarantee (postconditions): A user profile has been created and saved.
Main Success Scenario (or Basic Flow):

1. Mother selects “Create new profile” from the menu.
2. System presents an option, in the form of a pop-up, for typing in a name for the new profile.
3. Mother enters the name of the profile and clicks “Save”.
4. The system closes the pop-up.
5. The system saves the profile to stable memory.

Extensions (or Alternative Flows):
*a. At any time, System fails:
If the system fails and the program shuts down then only the saved data will be available. If the student hasn’t chosen “Save” in the pop-up, then the information is lost and a new profile has not been created.
3a. The name of the profile already exists.

1. System presents option to either overwrite the existing name, or create a new one.
    a. The mother chooses not to overwrite the profile.
        i. The mother chooses “No”.
        ii. The dialog is closed.
        iii. The mother enters another name for the profile and clicks “Save”.
    b. The mother chooses to overwrite the profile.
        i. The mother chooses “Yes”.
        ii. The dialog is closed.
        iii. The information saved on the previous profile is lost.

Use Case UC3 – Create budget

Stakeholder & primary actor: Mother of family: Wants to create a budget for her family.
Preconditions: The program is open. There are one or more known incomes and expenses. A profile has been selected.
Success Guarantee (postconditions): Budget has been created within the active profile. Data has been saved.
Main Success Scenario (or Basic Flow):

1. Mother selects “Create new budget” from the menu.
2. System presents an option, in the form of a pop-up, for typing in a name for the new budget.
3. System presents a view for creating a new budget, which consists of fields to add income, expense categories and sums.
4. Mother chooses the time period for the budget.
5. Mother chooses whether to enter an income or an expense.
6. Mother enters an amount.
7. Mother chooses a category.
8. The mother clicks “Add”.
9. The budgeted economic event is added in the list
   Mother repeats step 5-9 until satisfied.
10. Mother clicks “Done”.
11. The budget window is closed.
12. System saves the data to the profile as a budget.

Extensions (or Alternative Flows):
*a. At any time, System fails:
If the system fails and the program shuts down then only the saved data will be available. If the student hasn’t chosen “Done” in the dialog, then the information is lost.
2a. The name of the budget already exists.

2. System presents option to either overwrite the existing name, or create a new one.
    a. The mother chooses not to overwrite the budget.
        i. The mother chooses “No”.
        ii. The dialog is closed.
        iii. The mother enters another name for the budget and clicks “Save”.
    b. The mother chooses to overwrite the budget.
        i. The mother chooses “Yes”.
        ii. The dialog is closed.
        iii. The information saved on the previous budget is lost.

6a. Mother inputs a negative amount of the economic event.

1. The system rejects the amount and registers the absolute value.

7a. The category that the user wants to use does not exist.

3. The user adds another category.
4. The user chooses this category for the economic event.

10a. The sum of the incomes doesn’t cover the budgeted expenses.

1. The system alerts the user of the difference with a pop-up.
    a. The mother chooses to ignore this fact and presses “Ok”.
        i. The pop-up is closed.
    b. The mother chooses to alter the budget and presses “Cancel”.
        i. The pop-up is closed.

ii. The mother changes the expenses that are shown in the list.
iii. The mother clicks done.

Use Case UC4 – Compare budget with cash flow

Stakeholder & primary actor: Father of family: wants to compare a budget with the months cash flow.
Preconditions: Computer is on and program is started. A budget has already been defined and the expenses and incomes are inputted. All data has been saved to the father's profile.
Success Guarantee (postconditions): The father has been able to compare the cash flow from the previous month with the budgeted amounts.
Main Success Scenario (or Basic Flow):

1. The father chooses his profile.
2. The father chooses to analyse the cash flow.
3. A dialog is shown by the system.
4. The father chooses the budget that he wants to compare with.
5. The father chooses the time period to analyse.
6. The father chooses "Done".
7. The dialog is closed.
8. A textual overview of the cash flow is shown as a table. The amounts are compared with the chosen budget and the difference is shown.
9. The father chooses a graphical presentation.
10. The system opens a new window for the graphical presentation.
11. The system shows the budget and the actual cash flow for each category in a diagram.
12. The system also shows the budgeted and actual income.
13. The total sum is shown.
14. The father closes the window.
15. The father exits the program.

Extensions (or Alternative Flows):

5a. The father doesn't choose a time period.

1. The system uses the default value that is the previous month.

6a. The father changes his mind and doesn't want to analyse his economy.

1. The father chooses "Cancel".
2. The dialog is closed.

14a. The father wants to compare the existing diagram with another time period.

1. The father changes the time period in the textual presentation.
2. A new presentation is shown by the system.
3. The father chooses a graphical presentation.
4. The system opens another window for the second graphical presentation.
5. The father compares the two time periods.
6. The father closes the window.

14b. The father wants to compare the time period with another budget.

1. The father changes the budget in the textual presentation.
2. A new presentation is shown by the system.
3. The father chooses a graphical presentation.
4. The system opens another window for the second graphical presentation.
5. The father compares the two time periods.
6. The father closes the window.