Account This!

Group 9

Kristoffer Renholm
Johannes Edelstam
Joakim Ekberg
Jesper Skoglund

Group 9
Mjukvarukonstruktion
Use Cases and Requirements

Use Cases

Primary actor: the customer

Users

Register new company

Main scenario

1. The user is presented with a form for the company and the initial user.

2. The user fills out the form and enters all the required information and presses the register button.

3. If all the requested information is present the user is presented with confirmation screen that asks if all the entered information is correct. On this page there is two links. One that confirms the user's information and registers the company and the user. And one that takes the user back to the form to change some of the information. The user clicks the register link.

4. The system registers the company, the initial user and creates a new fiscal year that belongs to the company. That fiscal year is set to the current fiscal year. A mail is mailed to the initial user with their login information.

Alternative Scenarios

From 2:

1. If some of the information is missing the user sees the form again with information about which information that are missing or which information that has not meet the validation (like an email address that is not an email address, or that the email the user tries to use is already in the system, etc..).

2. The user corrects his mistakes and presses register again. If the validation passes he will be in the main scenarios step nr 3.

User login

Main scenario

1. The user is presented with a login screen asking the user to fill out her username and password.

2. The user fills out the required credentials and presses the login button or enter.

3. The system validates the username together with the password against the user-database.

4. If the system considers the user valid it forwards the user to the start page inside the application.

Alternative Scenarios:

From 3:

1. If the system considers that the username or password is incorrect the user is redirected back to the login page with an error message that tells the user such.

User logout

Main scenario

1. The user always has the ability to logout using the logout link.

2. If the user wants to logout he just clicks the link.

3. When the link is clicked the system cleans out the way the system keep track of that the user is logged in (i.e. a session variable or a cookie).

4. The system forwards the user to the main page with a message about that the user is now logged out.

5. The user reads the message about that he is now logged out.

Update user information

Main Scenario

1. The user navigates to the page that shows his information; it's a link on this page that will take the user to the page where he could edit his information.

2. If the user clicks this link the system verifies that the user is actually the user that the information belongs to. If so the user is forwarded to the page where he can update his information.

3. The user updates his information as he wants and when he is finished he clicks a button to save the changes.

4. The system once again verifies that the user actually is the user that the information belongs and validates the new information just like the information was validated upon registration. If the user is the required used and the information passes validation the information is saved and the user is forwarded to the pages that shows the updated information with a message that tells the user so.

Alternative Scenarios

From 2:

1. If the verification of the user fails but the user is an admin user he will be able to proceed and edit the user's information anyway. This is valid for step 4 in the main

scenario as well. The admin user should be able to save and update every user's information in the system.

From 2:

1. If the user isn't the user that the information belongs to he will be presented a page that tells him so.

From 3:

1. If the users tries to hack the system and submits data in a malicious way the user will be presented a page that this is now allowed.

Companies

Create a new user in an company

Main Scenario

1. The user navigates to its company page. On this page there is a link to another page where the user can add new users to its company. The user presses this link and the system redirects the user to this page.

2. On this page the users fills out an email to the new user. When this is done the user clicks the link to add the new user.

3. When the link is clicked the system validates that the email address actually is an email address. If the validation passes the system adds a temporary user and sends a mail to the new user asking him to fill out the rest of his information. Providing him with an URL to the page where he can do so.

4. The new user gets the mail and clicks the link.

5. The system forwards the user to the page where he can fill out the rest of the information required for the system. When the new user is done he clicks save.

6. Now the system validates his information and if the validates passes the user is added to the system under the right company.

Alternative Scenarios

From 3:

1. If the validation fails the user is redirected back to the page where he fills out the email address with a message that tells him that the validation failed.

From 6:

1. If the validation fails the new user is redirected back to the page where he fills out the information about himself with a message that tells him that the validation has failed.

Update the company information

Main Scenario

1. The user navigates to the page that shows the company information; it's a link on this page that will take the user to the page where he could edit the company information.

2. If the user clicks this link the system verifies that the user is actually a user that has access to the company information. If so the user is forwarded to the page where he can update the company information.

3. The user updates the information as he wants and when he is finished he clicks a button to save the changes.

4. The system once again verifies that the user actually is a user that has access to the company and validates the new information just like the information was validated upon registration. If the user is the required user and the information passes validation the information is saved and the user is forwarded to the page that shows the updated information with a message that tells the user so.

Alternative Scenarios

From 2:

1. If the verification of the user fails but the user is an admin user he will be able to proceed and edit the company information anyway. This is valid for step 4 in the main scenario as well. The administrator should be able to save and update every company information in the system.

From 2:

1. If the user doesn't have access to the company he will be presented a page that tells him so.

From 3:

1. If the users tries to hack the system and submits data in a malicious way the user will be presented a page that this is now allowed.

Fiscal year

Change fiscal year

Main scenario

1. The user navigates to the fiscal year page.

2. The user clicks on the link 'Start new fiscal year'

3. A confirmation telling the user that this will make all the vouchers and voucher rows from current fiscal year unable to edit shows up.

2. The user clicks 'Confirm'.
3. The system locks all vouchers belonging to the current fiscal year and marks the new fiscal year as current.

Alternative scenario

From 2:

1. The user clicks 'Abort'
2. The system doesn't make any changes to the current fiscal year.

Vouchers

All actions that change the state of an earlier created voucher is only allowed if the voucher belongs to the current fiscal year.

Create a new voucher

Main scenario

1. The user navigates to the vouchers page.
2. The user selects 'New voucher'.
3. The user sees a view where new voucher rows can be added.
4. Selects **Create a new voucher row** multiple times.
5. The system saves the voucher and set the vouchers fiscal year to the company's current fiscal year.
6. The user returns to the vouchers page.

Alternative scenarios

From 5:

1. If the voucher contains zero rows, the user will be warned and the voucher will not be saved.

Remove a saved voucher

Main scenario

1. The user navigates to the vouchers page.
2. The user clicks on the link named 'Mark as removed' beside the voucher in question.
3. The selected voucher is marked as removed.

Edit a saved voucher

Main scenario

1. The user navigates to the vouchers page.
2. The user clicks on the link named 'Create replacement voucher'.
3. The selected voucher is marked as removed.
4. A new voucher is created which is linked to the old one, but marked as a replacement for the old one.
5. The scenario continues as from step 3. in the **Create a new voucher** main scenario.

Voucher rows

All changes that apply to earlier saved voucher rows require that the voucher rows voucher belongs to the current fiscal year.

Create a new voucher row

Main scenario

1. The user selects 'New voucher row'.
2. The user fills in the sum and chooses an account for this row.
3. The user chooses weather it is a debit or credit row.
4. The user saves the row.

Alternative scenarios

From 4:

1. If the voucher row doesn't have an account or the sum is 0 the user will be warned and the voucher row will not be saved.

Edit a saved voucher row

Main scenario

1. The user navigates to the vouchers page.
2. The user clicks the link 'Edit voucher'.
3. The user is presented with all the voucher rows that belong to the selected voucher.
4. Each voucher row has a link with the text 'Replace' next to it.
5. The user clicks 'Replace' next to one of the voucher rows.
6. The selected voucher row is marked as removed.

7. A new voucher row appears marked as a replacement to the old.

8. The scenario continues as from step 2. in the **Create voucher row** main scenario.

Remove a saved voucher row

Main scenario

1. The user navigates to the vouchers page.

2. The user clicks the link 'Edit voucher'.

3. The user is presented with all the voucher rows that belong to the selected voucher.

4. Each voucher row has a link with the text 'Mark as removed' next to it.

5. The user clicks 'Mark as removed' next to one of the voucher rows.

6. The selected voucher row is marked as removed.

Accounting plan

Create accounting plan

Main scenario

1. The user navigates to the accounting plan page.

2. The user selects ‘Create accounting plan’.

3. The user is presented with a text box and a submit button.

4. The user enters the name of the accounting plan in the text box.

5. The user presses the submit button.

6. The system creates a new accounting plan.

7. The system redirects the user to the accounting plan edit page.

Alternative scenario

From 5:

1. If the text box is empty, the user will be warned and the accounting plan will not be saved.

Duplicating an accounting plan

Main scenario

1. The user navigates to the accounting plan page.

2. The user selects an accounting plan and then clicks ‘Duplicate accounting plan’.

3. The user is presented with a text box, selection drop-down and a submit button.
4. The user enters the new name of the accounting plan in the text box.
5. The user chooses from the selection drop-down to which company the accounting plan will be copied.
6. The user press the submit button.
7. The system copies the accounting plan.
8. The system redirects the user to the new accounting plan edit page.

Alternative scenario

From 6:

1. If the text box is empty, the user will be warned and the accounting plan will not be copied.

From 6:

1. If the there already exists an accounting plan with the named specified in the text box for the company chosen in the selection drop-down, the user will be warned and the accounting plan will not be copied.

Deleting an accounting plan

Main scenario

1. The user navigates to the accounting plan page.
2. The user selects an accounting plan and then clicks ‘Delete accounting plan’.
3. The user is presented with a confirmation prompt that asks if the user is sure about deleting the accounting plan.
4. The user chooses ‘Yes’
5. The system removes the accounting plan.
6. The system redirects the user to the accounting plan page

Alternative scenario

From 3:

1. If the accounting plan is in use in any fiscal year, the system will deny the user from deleting the accounting plan.

From 4:

1. If the user has chosen ‘No’ in the confirmation prompt the accounting plan will not be deleted.

Adding accounts to an accounting plan

Main scenario

1. The user navigates to the accounting plan page.
2. The user selects an accounting plan and then selects 'Edit'.
3. The user selects 'New account'
4. The user is presented with a form for the new account.
5. The user fills out the form.
6. The user press the submit button.
7. The system saves the new account to the accounting plan.

Alternative scenario

From 6:

1. If an account with the same account number already exists in the accounting plan the user is warned and must choose another account number.

From 6:

1. If one of the required fields is not filled in the system warns the user and the new account will not be saved.

Removing accounts from an accounting plan

Main scenario

1. The user navigates to the accounting plan page.
2. The user selects an accounting plan and then selects 'Edit'.
3. The user selects an account and then selects 'Delete account'
4. The user is presented with a prompt asking if the user is sure about deleting the account.
5. The user selects 'Yes'
6. The system removes the account from the accounting plan.

Alternative scenario

From 5:

1. If the user selects 'No', the account will not be removed.

From 5:

1. If the account to be removed is in use in any voucher the user will be denied deleting the account.

Support

Main scenario

1. The user selects 'Help' from the menu.
2. The user is presented with a text box and a submit button.
3. The user writes her question in the text box.
4. The user clicks the submit button.

Alternative scenarios

From 4:

1. If the text box is empty, the user will be prompted to write a question, otherwise the form will not be submitted.

Requirements

Non-functional

Unlike the functional requirements used to specify the exact behavior of Account This!, the non-functional requirements have first and foremost been mentioned to better depict the constraints and criteria that can be used to judge the general functionality of the system (Sommerville 2007, pp. 119). The non-functional – whether related to the product, the organization or fully external – are not directly linked with any individual system features but rather with the system itself (ibid., pp. 121-123). From the project description, it's evident that the intended customers (mainly consisting of people working in small firms) will desire security and reliability above anything else. Account This! must also come with a high degree of usability, making the program intuitive and easy to use. A fourth major aspect category is standards. These categories alone however won't sell the product; several other categories of non-functional requirements are relevant as well.

Security

1. The Account This! system shall be able to recoil from any unwanted break-in attempts.
2. System administrators shall be able to make daily backups of user data to protect its users from any loss of information.
3. Accounts and user information shall be ciphered and unavailable to all – including skilled hackers – except to the users themselves and to the system administrators.
4. 90 % of the system caused errors should be possible to detect through a comparison between the logged actions/events and the present outcome.

Usability

1. The layout of the user interface, such as the general journal and nominal ledger, must be customizable to better fit the variety of customer needs.
2. Users shall be able to grasp the general idea of the system in less than 3 minutes.
3. Users shall be able to learn how to use the system's functionality in less than 15 minutes, given that they have some previous experience of bookkeeping.
4. The website encompassing Account This! must be easy to navigate. The rate of hyperlink mouse clicks leading to a page not intended must be less than 3 %.
5. No unnecessary pages or boxes shall hinder the efficient use of Account This!. Each function shall consist of no more steps than would have been necessary with traditional pencil and paper bookkeeping.
6. A hyperlink to a help webpage containing sentences with explanatory text shall be accessible at all time.
7. 99 % of the elements in Account This! shall be intuitive or clarified with clear text.
8. Each and every action should be logged and stored by the system, granting users an opportunity to view a history of his/her past changes and events.

Reliability

1. System failures caused by functions behaving different than they're supposed to must be kept low, at a failure rate occurrence of 2 %.
2. Account This! shall be able to detect and explain at least 95 % of the errors that may occur (by showing a brief text message).
3. The probability of data corruption on failure must be less than 1 %.

Standards

1. All functions embedded in the bookkeeping system must meet all requirements imposed by Swedish law.
2. Account This! shall support all elements of the BAS accounting plan.
3. Each user must be able to change, save and use whatever standardized date format he or she prefers. Information concerning the chosen format must always appear in connection to any listed dates.
4. Each company must be able to change, save and use any existing currency format they desire. Information concerning the chosen format must always appear in connection to any listed value.
5. All corporate trademarks mentioned in the Account This! system must be written in accordance with Swedish laws and regulations.

Availability

1. The probability rate of function unavailability must be less than 0.5 % of the total system uptime.
2. The probability rate of webpage unavailability must be less than 1 % of the total system uptime.
3. Account This! shall be available from any Swedish ISP connection.
4. The system must be able to use from all computers equipped with modern web-browsers. No additional software shall be required.

Performance

1. The system must be built to keep loading times as short as possible.
2. The non-bandwidth consuming user/event response time must be imperceptible.

Scalability/extensibility

1. The time or effort required to implement changes in the software shall consist of no more than the energy that was put in to build the modification. Implementation of adjustments must consequently be straightforward a là copy and paste.
2. Each method in Account this! shall be well-documented to ensure that the system is both easy to understand and expand.

Quality

1. All values shown in the system must be 99.9 % accurate to the numbers once entered.

Compability

1. The system shall support all operating systems that are compatible for, and equipped with, any of the modern web-browsers the system is designed for: Firefox 2.0, Internet Explorer 7 and Safari 3.

Functional

Users

1. The user should be able to register a new company in the system providing information about the company and the initial user.
    1. The required information about the company is
        1. Name
        2. Company Form
        3. Organization Number
        4. Address (Standard Swedish address)
    2. The required information about the initial user is
        1. Name
        2. Address (Standard Swedish address)
        3. Phone-number
        4. Email
2. The user should be able to login using their username and password.
3. The user should be able to logout.
4. The user should be able to update their information.

Companies

1. The company could have several users connected to the company.
2. Each user in a company could easily create a new user in the company and then the system mails the login information to the new user.
3. Each user in a company should be able to update the company's information.

Fiscal years

1. Each company must have at least one fiscal year.
2. A company should not be able to have more than one current fiscal year.
3. Once a fiscal year is closed, it should not be able to be reopened.

Vouchers

1. All actions made on vouchers should be date and time stamped.
2. Users should be able to create vouchers
3. A user should only be able see and edit vouchers belonging to his or hers company.
4. Once a voucher is saved, a voucher should never be able to remove.

5. Users should be able to create replacement vouchers to already existing vouchers it the voucher belongs to the current fiscal year. A replacement voucher will then act as the new voucher.
6. Users should be able to strike out vouchers belonging to the current fiscal year. They will not be removed, but marked as removed.
7. To each non saved voucher belonging to the current fiscal year, users should be able to create/update/remove vouchers rows.
8. To each saved voucher belonging to the current fiscal year, users should be able to remove vouchers rows.
9. Each voucher row belongs to an account and a fiscal year.
10. Each voucher row has a sum that is greater or less than zero. If it is greater than zero the voucher row is a credit row, otherwise it is a debit row.

Accounting plan

An accounting plan is a set of accounts used in double entry bookkeeping.

1. Users should be able to create new accounting plans.
2. Users can only edit accounting plans associated with one of the companies associated with the user.
3. An accounting plan is created for and belongs to one and only one company, except shared accounting plans, see 6.
4. The user shall be able to duplicate accounting plans across companies associated with the user.
5. Accounting plans can be removed when not used as an accounting plan in any fiscal year.
6. Administrators shall be able to template for accounting plans shared across all companies.
7. An account belongs to one and only one accounting plan (the same account number can however be used in multiple accounting plans).
8. Users should be able to edit accounts if not used in any voucher.
9. Users should be able to add accounts to an accounting plan.
10. Accounts not used from any voucher can be removed from an accounting plan.

Support

Users shall be able to send questions in an easy way to a support staff

1. Questions are sent as a e-mail to a configurable e-mail address
2. The only input required by the user is the question itself
3. Name and e-mail address to respond to shall be automatically, and transparent for the user, be provided from the user's profile
4. A copy of the question shall be sent to the user's e-mail address

References

Sommerville, I. (2007). Software Engineering, Eight Edition, Pearson Education, Essex, England.