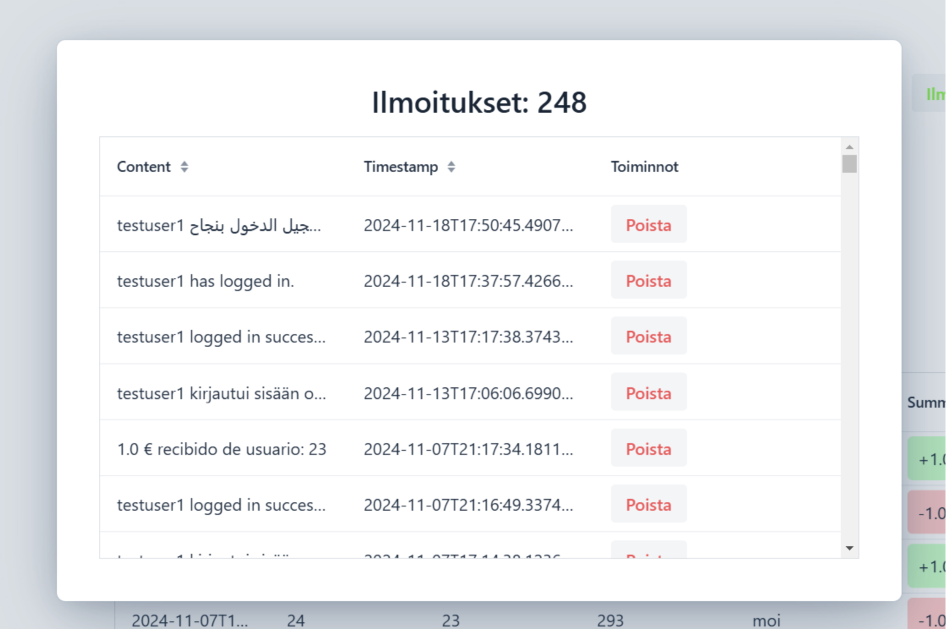
Six Sigma Report 1: Notification Grid Localization Issues



Ilmoitukset: 248

Content	Timestamp	Toiminnot
testuser1 جيل الدخول بنجاح...	2024-11-18T17:50:45.4907...	Poista
testuser1 has logged in.	2024-11-18T17:37:57.4266...	Poista
testuser1 logged in succes...	2024-11-13T17:17:38.3743...	Poista
testuser1 kirjautui sisään o...	2024-11-13T17:06:06.6990...	Poista
1.0 € recibido de usuario: 23	2024-11-07T21:17:34.1811...	Poista
testuser1 logged in succes...	2024-11-07T21:16:49.3374...	Poista

“Content” and “Timestamp”- titles are not localized. Screenshot is taken from notification grid component on Thursday 14.11.

1. Define: Understand the Problem and Set Clear Objectives

- **Objective:**

  Improve the notification grid by localizing its titles and ensuring they are properly translated across all supported languages.

- **Goal:**

  Localize all notification grid titles within the sprint, leading to improved user experience.

- **Problem Statement:**

  "The notification grid titles are not localized in any language, affecting usability for non-English-speaking users."

2. Measure: Collect Data to Understand the Current State

- **Baseline Metrics:**
  - Number of untranslated notification grid component titles: 66,6% of the titles.
  - Issues related to notification grid localization found during functional testing: 2 issues.
- **Data Collection:**
  - Track untranslated notification grid titles.
  - Document any delays or errors in the display of localized content.

3. Analyze: Identify Root Causes of Defects

- **Root Cause Analysis:**
  - The notification grid titles were not included in the localization process, leading to missing translations.
- **Root Cause Analysis (5 Whys):**
  - Why are the notification grid titles not localized? → Because the titles were not added to the localization.
  - Why were they not added? → Because the localization process was not comprehensive and did not include notification grid elements.
  - Why was the process incomplete? → Because notification grid localization was planned for a later phase.
  - Why was it delayed? → Due to prioritization of core functionalities over localization.
  - Why was localization not prioritized? → Lack of coordination between development and localization teams.

4. Improve: Develop and Implement Solutions

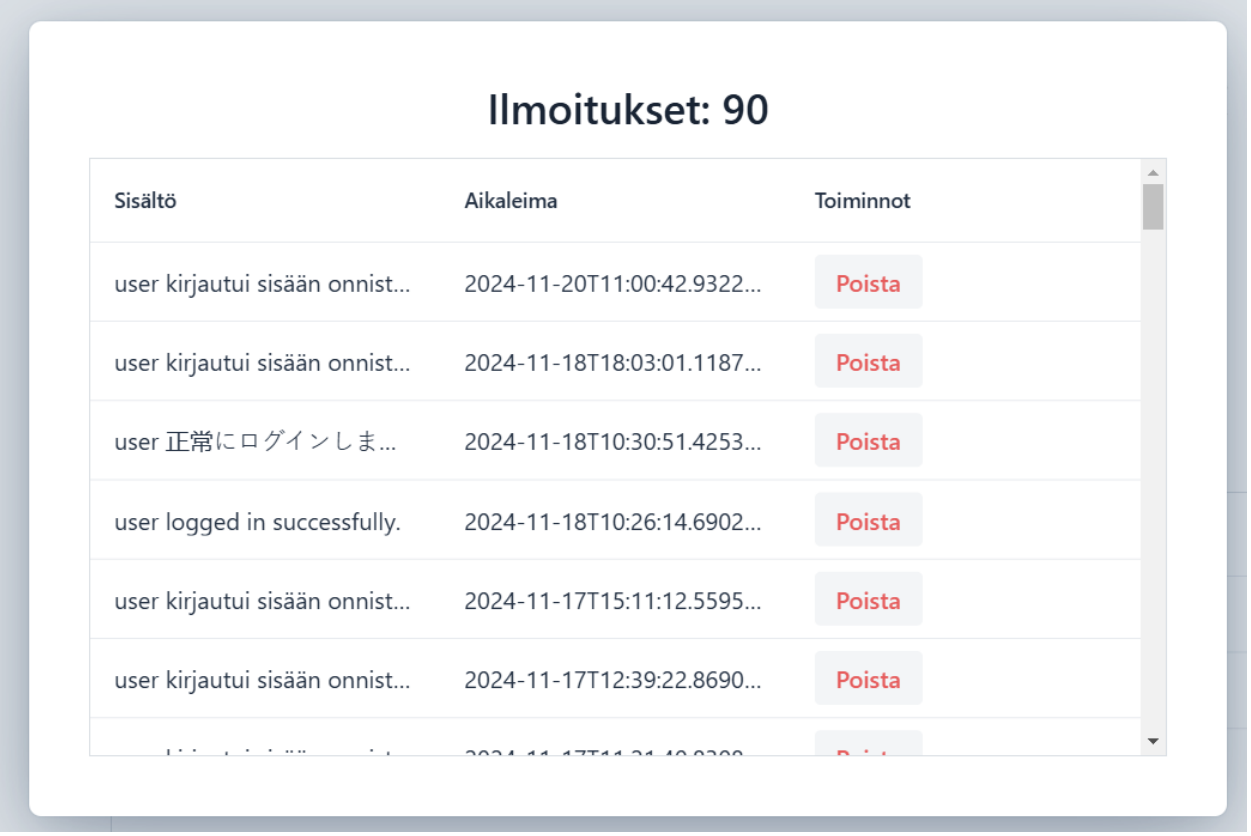
- **Solution Development:**
  - Include notification grid titles in the localization
  - Use a translation tool to ensure consistent and complete localization.
  - Update language bundles to include translations for titles.

- **Solutions:**
    - Use, for example, google translate, to ensure all necessary translations are correct.
    - Update language bundles to include translations for titles.
    - Make code changes in main view to ensure notification grid titles are localized correctly.
    - Conduct a comprehensive review and quality testing after localization.

5. Control: Ensure the Improvements Are Sustained

- **Monitoring:**
    - Set up automated tests to detect untranslated notification grid titles.
    - Set up JUnit tests for language bundles to ensure correctness of translation and localization.
- **Documentation and Continuous Improvement:**
    - Document the new localization process for notification grid titles.
- **Example Control Plan:**
    - **Automated Testing:** Implement automated checks for untranslated strings in the notification grid.
    - **Feedback Loop:** Establish a feedback loop to collect and address localization issues raised by users.

Notification grid component after applying six sigma principles:

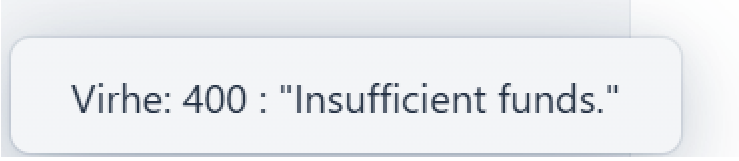


Ilmoitukset: 90

Sisältö	Aikaleima	Toiminnot
user kirjautui sisään onnist...	2024-11-20T11:00:42.9322...	Poista
user kirjautui sisään onnist...	2024-11-18T18:03:01.1187...	Poista
user 正常にログインしま...	2024-11-18T10:30:51.4253...	Poista
user logged in successfully.	2024-11-18T10:26:14.6902...	Poista
user kirjautui sisään onnist...	2024-11-17T15:11:12.5595...	Poista
user kirjautui sisään onnist...	2024-11-17T12:39:22.8690...	Poista

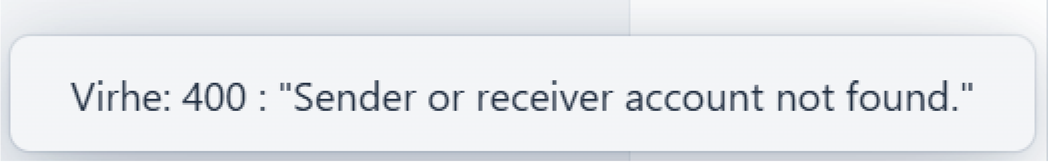
As you can see, all the titles are now correctly localized in the notification grid component and are now available in all supported languages. The screenshot was taken on Wednesday 20.11 from the notification grid.

Six Sigma Report 2: Transaction Error Handling Localization



Virhe: 400 : "Insufficient funds."

Error message for transaction error: “insufficient funds” is not localized in any language. The screenshot was taken from the main view on Thursday 14.11 from the main view.



Virhe: 400 : "Sender or receiver account not found."

Error message for transaction error: “Sender or receiver account not found.” is not localized in any language. The screenshot was taken from the main view on Thursday 14.11 from the main view.

1. Define: Understand the Problem and Set Clear Objectives

- **Objective:**

  Ensure that error messages related to transaction issues are localized and accurately communicated.

- **Goal:**

  Localize all transaction-related error messages within the sprint and reduce the number of complaints about transaction errors.

- **Problem Statement:**

  Transaction error messages such as 'Insufficient funds' and 'Receiver not found' are not localized, causing confusion for non-English-speaking users.

2. Measure: Collect Data to Understand the Current State

- **Baseline Metrics:**
  - Per cent of untranslated transaction error messages: 100%.
  - Issues related to transaction localization found during functional testing: 2
- **Example Data Collection:**

    - Track all transaction error messages and their localization status.
    - Review error logs to determine the frequency of missing localization for transaction-related errors.

3. Analyze: Identify Root Causes of Defects

- **Root Cause Analysis:**
    - The error messages were hardcoded, not included in the localization framework.
    - The localization process was incomplete and not fully communicating with the language bundles.
- **Example Root Cause Analysis (5 Whys):**
    - Why are the error messages not localized? → Because the error messages were hardcoded.
    - Why were they hardcoded? → The app was originally not built for localization.
    - Why was localization omitted? → Forgotten during the translation process.
    - Why was there no process? → The amount of data was massive to translate, naturally some cases were overlooked by accident.
    - Why was this coordination lacking? → Poor management

4. Improve: Develop and Implement Solutions

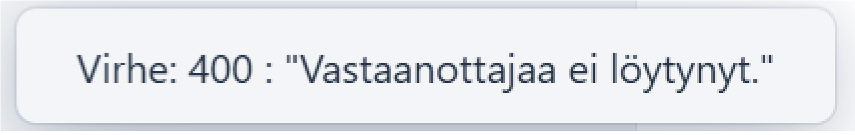
- **Solution Development:**
    - Refactor transaction error messages to be part of the localization system.
    - Work on localization to ensure error messages are translated into all supported languages.
- **Example Solutions:**
    - Refactor code to include transaction errors in the localization system.
    - Update language bundles to include translations in all supported languages.
    - Confirm correct localization using translation tool.

5. Control: Ensure the Improvements Are Sustained

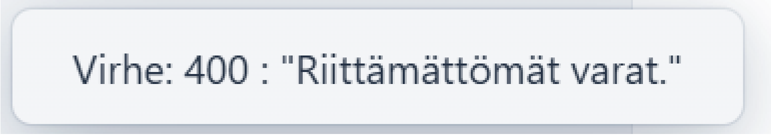
- **Monitoring:**

    - Automated tests to identify any untranslated transaction-related error messages.
    - Track complaints related to transaction errors after localization.
- **Documentation and Continuous Improvement:**
    - Document the new error handling and localization process for transactions.
    - Continue to make iterative improvements as necessary.
- **Example Control Plan:**
    - **Automated Testing:** Implement automated checks to ensure all transaction-related error messages are localized.
    - **Feedback Loop:** Establish a process to collect user feedback on the clarity and accuracy of error messages.

Transaction related errors after applying six sigma principles:



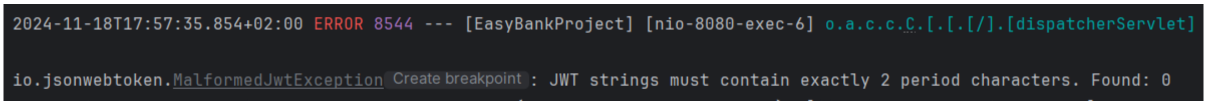
Virhe: 400 : "Vastaanottajaa ei löytynyt."



Virhe: 400 : "Riittämättömät varat."

Transaction related errors are now localized in all supported languages. The screenshots were taken on Wednesday 20.11 from the main view.

Six Sigma Report 3: Authentication and Logout Issues



```
2024-11-18T17:57:35.854+02:00 ERROR 8544 --- [EasyBankProject] [nio-8080-exec-6] o.a.c.c.C.[.[.[/].[dispatcherServlet]

io.jsonwebtoken.MalformedJwtException Create breakpoint : JWT strings must contain exactly 2 period characters. Found: 0
```

When user logs out, "Malformed JWT exception"- error occurs. The screenshot was taken from the console on Friday 15.11.2024.

1. Define: Understand the Problem and Set Clear Objectives

- **Objective:**

  Resolve the "Malformed JWT exception" error that occurs during logout and improve the logout process to ensure a smooth user experience.

- **Goal:**

  Fix the logout error within the sprint and ensure that users can log out without issues.

- **Problem Statement:**

  "Malformed JWT exception' error occurs when users attempt to log out, disrupting the user experience."

2. Measure: Collect Data to Understand the Current State

- **Baseline Metrics:**
  Frequency of the "Malformed JWT exception" error: every time user logs out.
- **Data Collection:**
  - Analyze server logs to track the frequency and conditions under which the error occurs.

3. Analyze: Identify Root Causes of Defects

- **Root Cause Analysis:**
  - The JWT token is being invalidated while the ongoing session still expects valid token.
  - Lack of error handling in the JWT token validation and UI authentication processes, leading to the exception.
- **Root Cause Analysis (5 Whys):**
  - Why does the "Malformed JWT exception" error occur? → Because current session expects valid token, so it throws exception.
  - Why does the exception cause error? → Because the exception is not handled the right way.

  - Why isn’t the exception handled? → Complete error testing and handling had not been done.
  - Why were error handling and testing not done? → Since the JWT exception didn’t crash the app immediately, the task was postponed.
  - Why was it not tested? → The focus was on core authentication functionality, not error scenarios.

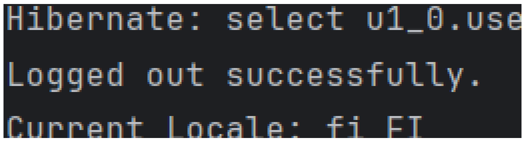
4. Improve: Develop and Implement Solutions

- **Solution Development:**
  - Make sure exceptions are thrown and caught properly in JWT validation and UI authentication processes.
  - Test the logout process thoroughly to ensure no errors occur.
- **Solutions:**
  - Make the JWT validator function to throw an exception if the token is null.
  - Make sure that the “Before Enter” functions, which authenticate the UI, are catching the exceptions properly that are thrown by the JWT validator.
  - Perform end-to-end testing of the logout function across all environments.

5. Control: Ensure the Improvements Are Sustained

- **Monitoring:**
  - Monitor server logs for JWT-related errors after implementation.
- **Documentation and Continuous Improvement:**
  - Document the updated logout process and error handling strategies.
  - Continuously collect feedback and refine the process.
- **Example Control Plan:**
  - **Automated Testing:** Add automated tests to ensure the logout function and UI authentication works as expected.
  - **Feedback Loop:** Collect user feedback after the fix to ensure the issue is fully resolved.

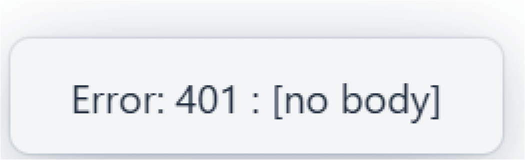
Log out function output after applying six sigma principles:



```
Hibernate: select u1_0.use
Logged out successfully.
Current Locale: fi_FI
```

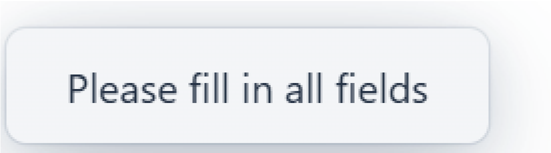
When user logs out, “Logged out successfully.”- message is logged in the console and user is redirected successfully in the login page and won’t be able to access other pages anymore without logging in or registration. The screenshot was taken on Wednesday 20.11.

Six Sigma Report 4: Registration and Login Error Message Localization



Error: 401 : [no body]

“Error: 401: [no body]”- occurs when user does not fill in all the required fields in login or the credentials are invalid. This also appears when the user tries to register with an already existing username. The screenshot was taken on Thursday 14.11 from log in view.



Please fill in all fields

“Please fill in all fields”- error message is not localized in any language. This should occur when the user tries to log in or register without filling in the username or password. The screenshot was taken on Thursday 14.11 from register view.

1. Define: Understand the Problem and Set Clear Objectives

- **Objective:**

  Localize all error messages related to registration and login, ensuring users can understand and resolve issues. Also make sure to implement valid error handling.

- **Goal:**

  Localize all error messages related to registration and login within the sprint and improve user experience and handle all unhandled errors.

- **Problem Statement:**

  "Error messages indicating problem within registration or login are not localized or handled, causing confusion for non-English-speaking users."

2. Measure: Collect Data to Understand the Current State

- **Baseline Metrics:**
  - Number of untranslated registration and login error messages: 4
  - Number of unhandled error messages during registration and login processes: 3
- **Data Collection:**
  - Track all error messages during registration and login processes.
  - Analyze server logs to track the frequency and conditions under which the error occurs.

3. Analyze: Identify Root Causes of Defects

- **Root Cause Analysis:**
  - Error messages were neither tracked nor processed.
  - Localization was completely missing.
- **Cause Analysis (5 Whys):**
  - Why are the error messages not localized? → Messages were originally hardcoded.
  - Why were they hardcoded? → Team saw no need for custom error handling early phase when these were created.
  - Why are there unhandled errors? → The focus was on core functionality, not error messages.
  - Why was the focus on functionality? → Due to strict timetable functional development was prioritized.
  - Why were localization and error handling not prioritized? → Poor vision of the full frame.

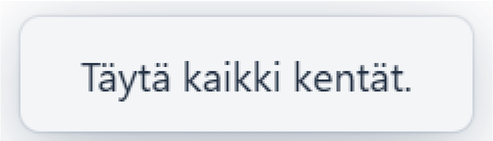
4. Improve: Develop and Implement Solutions

- **Solution Development:**
  - Refactor error messages to ensure they are dynamic and part of the localization framework.
  - Translate all error messages related to registration and login.
  - Conduct thorough testing to ensure accurate translations.
- **Solutions:**
  - Move all error messages to the localization framework --> update language bundles and implement them to login and register UI's.
  - Make sure controllers throw valid exceptions and catch them in the UI with correctly localized error messages.
  - Test for clarity and ensure messages are user-friendly.

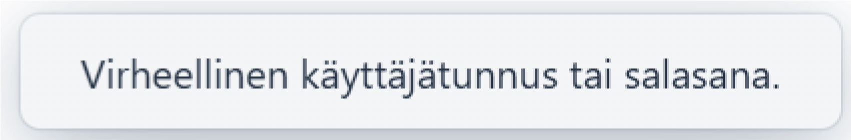
5. Control: Ensure the Improvements Are Sustained

- **Monitoring:**
  - Set up automated tests to detect untranslated error messages.
  - Monitor issues regarding registration and login errors.
- **Documentation and Continuous Improvement:**
  - Document the localization process for registration and login error messages.
  - Continuously review and improve the error messages
- **Example Control Plan:**
  - **Automated Testing:** Implement automated checks for untranslated messages.
  - **Feedback Loop:** Regularly gather user feedback on error message clarity and localization accuracy.

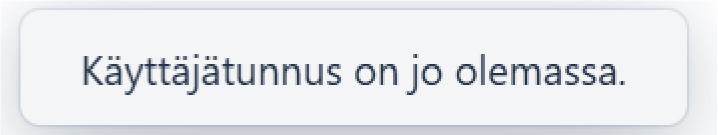
Log in and registration related error messages, after applying six sigma principles:



Täytä kaikki kentät.



Virheellinen käyttäjätunnus tai salasana.



Käyttäjätunnus on jo olemassa.

All the log-in and registration related errors are now handled successfully and are also localized in all supported languages. The screenshots were taken on 20.11 from log in and registration UI’s.

Reflection

The Six Sigma documentation was truly an eye-opening experience! We discovered loads of localization and error handling problems we weren’t aware of. Documentation helped us to track down the process to fix issues and prevent them from happening further on again. We are looking forward to using six sigma principles in further projects.