Business Requirements Document

Summary

Description of a new feature for the Internet banking application: the ability to hide accounts in the user's personal account. The feature allows the user to customize the interface by hiding irrelevant accounts, improving navigation and perception of the personal account The terms used in this document are provided in the SRS document for this feature.

Project goals

Goal	Metric	Target value	Deadline	Monitoring
Increase satisfaction	NPS of users who used hiding	+10% to the current level	3 months after release	Through surveys, in-app reviews, contact center
Reduce requests	Number of support tickets for displaying accounts	-15% of the current level	In 2 months	Through the ticket system/td>

Project history

Clients often open accounts for specific purposes: savings, transfers, purchases. Some become obsolete but are not closed. This leads to interface overload and complicates the perception of information. A large amount of information can be collected in the personal account in the application. The personal account looks cluttered and inconvenient to use.

Stakeholders

- Development team
- Product owner (PO) of the Internet banking application
- Retail products department (individuals)
- Customer support
- UX/UI team
- Infrastructure / DevOps
- QA, testing team

Time: according to the preliminary estimate of the development team, the implementation period is from 2 to 4 weeks at the current workload. The estimate is specified after the requirements are finalized.

Budget: the cost of the development team's work, testing, implementation within 1 month; cost of new required capacities

Project scope:

1. Implementation of the function of hiding accounts from the main screen of the Internet bank.
2. Ability to return the hidden account to display.
3. Displaying the hidden account in a separate list.

Limitations

1. The user does not have the ability to hide blocked accounts.
2. The user does not have the ability to hide the primary account.

3. The hidden account retains all the functionality of the active account, including transfers, replenishment and viewing the history of transactions, hiding does not mean blocking the account.

User Stories:

1. As a user with many outdated accounts, I want to hide them from the interface so I don't waste time scrolling through them and don't make a mistake when transferring.
2. As a user, I want to be able to view a hidden account so I don't lose access to the information if I need it again.
3. As a user, I want to see hidden accounts in a separate section so I can quickly restore the display if needed.

Use cases:

UC-1: Hiding an account
UC-2: Ability to view hidden accounts
UC-3: Ability to return a hidden account to the main screen (the account is displayed again in the general "My accounts" list)
UC-4: Error when trying to hide a primary / blocked account (When trying to hide a primary or blocked account, a notification appears with the text "Unable to hide the account" and an "OK" button.)
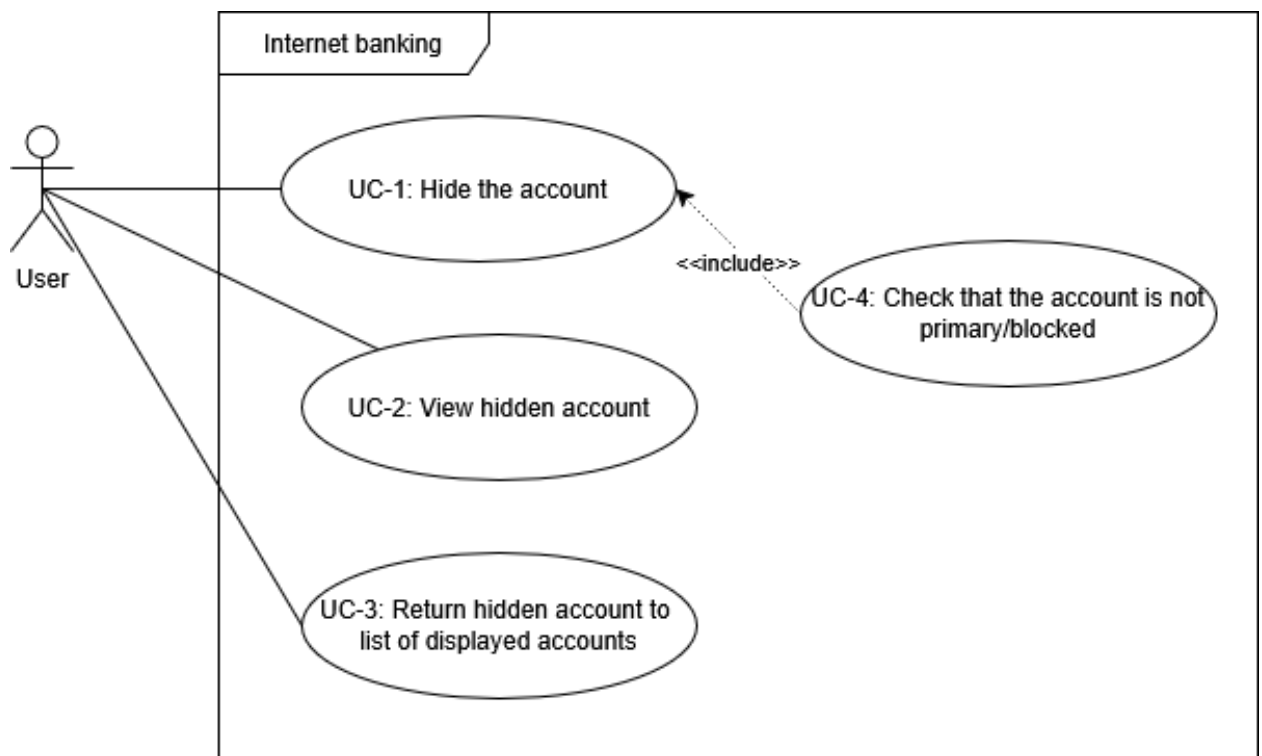
Product requirements

Functional requirements:

FR.1: Ability to hide the display of an account (except for the primary and blocked one) in the user's personal account in application

FR.2: Ability to return the display of a hidden account

Non-functional requirements

NFR.1: Hiding/returning the display of an account should not take longer than 1 second in the application. The account display (including hiding/returning) should be without noticeable delay with a weak internet connection and under high server load conditions.

NFR.2: Security: hiding the account does not affect access to transactions, there is no data leakage.

NFR.3: Cross-platform support: works on iOS and Android.

NFR.4: Reliability: The hiding state is saved when the application is restarted.

Acceptance criteria:

CR.1: The user can hide the account by performing no more than 3 actions: 1) click on the account card, 2) select "Hide account", 3) confirm.

CR.2: There is no data loss during the process of hiding the account.

CR.3: Hiding and restoring an account takes less than 1 second.

Release plan:

1. Step-by-step release plan

- Functionality testing: conducting testing, fixing bugs.
- Release to production: deploying an update to production, checking the functionality of the function.
- User updates: informing users through notifications about the new function.

2. Post-release monitoring

- Performance: using analytics to track response and load times.
- User feedback: analyzing feedback through the help desk and analytical tools.

3. Bug Support

- Error Handling: Identifying and prioritizing bugs via the bug tracking system. Fixing critical errors.
- User Support: Answering user questions via the support service, updating FAQ.