

Meeting Agenda

Group: OOPP-Group77

Date: 28/9

Chair: Hampus

Participants: David, Elin, Martin, Alexey, Hampus

Objectives (5 min)

- Decide how to do with ApplicationManager to make sure there is only one per runtime.
- Decide how to do with exceptions (where to handle them and how to write code with exceptions elsewhere).
- Discuss comments on code.
- Decide what to do with AccountHandler w.r.t. its currently too big a role and Elin's proposed change to a more clear model-control design.
- Discuss potential addition of sorting and searching functionality (starting with new user stories).

Reports (15 min) from previous meeting

Elin: Has sketched on an updated design model taking into the account received from Aline, and read up on dependency injection vs singleton pattern for the controller-ApplicationManager-part.

Hampus: Added parts to RAD (DoD), implemented and refined the method/code within LocalDiscStorage mainly, but also within other parts. For example, removed redundant text fields from the view.

David: Fixed ListViewItem and ListItemController but some things are currently missing in other parts of the application.

Martin: Implemented parsing emails and refreshing from server for Gmail.

Alexey: Fixed writingView and sending but cannot test due to active account part not being in place.

Discussion items (35 min)

- Decide how to do with ApplicationManager to make sure there is only one per runtime.
- Decide how to do with exceptions (where to handle them and how to write code with exceptions elsewhere)
- Discuss comments on code.

- Decide what to do with AccountHandler w.r.t. its currently too big a role and Elin's proposed change to a more clear model-control design.

Outcomes and assignments (5 min)

Outcomes of discussions:

- Decide how to do with ApplicationManager to make sure there is only one per runtime.
 - It was decided that:
 - We should ensure this by the use of dependency injection via the MainController to other controller's needing to access the ApplicationManager object. The other alternative would have been to use a singleton pattern.
- Decide how to do with exceptions (where to handle them and how to write code with exceptions elsewhere)
 - It was decided that:
 - By the new design (see below), with ApplicationManager as control, the ApplicationManager should have all try-catch-blocks and classes further down in the hierarchy should only throw exceptions without handling them, unless justified for the flow of that specific method.
- Discuss comments on code.
 - It was decided that:
 - Comment specific lines to make code easier to understand for everyone
 - Write a short summary of what the method does in javadoc and not only parameters and return value.
 - Add author to javadoc so it's easy for group to ask questions to the right person
 - Use descriptive method names to make it easier to understand what the method does.
- Decide what to do with AccountHandler w.r.t. its currently too big a role and Elin's proposed change to a more clear model-control design.
 - It was decided that:
 - Once the application is in a better shape (all parts are correctly working together) using the current design model, the application will be refactored into the new model proposed by Elin.

(re)Assign to each group member which issues to solve:

The group members were assigned the following tasks to do right now, and will get new tasks once their part is done:

David: Make emails readable, by clicking on them in the overview.

Martin: Connect controllers via dependency injection so that the application only has one ApplicationManager and thus one single state. Also fix so that folders are displayed and selectable, making the right emails show up.

Elin: Fix dropdown and make it possible to choose an account from multiple added accounts (Hampus does the storage part for this).

Hampus: Implement missing functionality in LocalDiscStorage, first and foremost retrieveAccount to help Elin.

Alexey: Finish implementing sending functionality.

Wrap up

- Unresolved issues for next meeting.
 - Testing remains an issue and will be brought up with Aline. Mainly, testing storage's I/O operations is challenging and difficult to do in a good way, due to circular dependencies in the tests between the tested methods.
- Decide a time and location for the next meeting.
 - 08:00 Thursday 30/9 at campus.