Flattie

Description:

The main requirements were to produce a software that is used by flat mates that are staying in the same flat. It's supposed to track groceries, finances, and chores of the apartment. Among the three software engineers the project was split into three parts, the first part is how the database is buffered, written, and read, the second part is how the user interacts with the database as well as its features, and finally the third part was connecting all this into a single GUI. Java Swing was used for the GUI.

What went well?

The three group members were able to communicate efficiently between each other as well as use various tools such as git, and github (version control tools).

What didn't go so well?

- Integration of the GUI with the database, and the functionality of the buttons. (Unfortunately, due to time constraints we developed the main parts into the console so there remains a software without GUI)
- Data type coordination between the classes was a bit complicated and intense, especially for the database integration.
- Reading the database from the references in the gradle container did not work properly.
- Testing different combinations of classes and databases was not working properly.
- There was a fail in the creation of the grocery and chore lists but during testing the other methods worked fine.
- When testing each method individually each method worked in the way it was intended but when trying to combine all some methods fail, so further testing and debugging needed
-

What have I learned?

- How to develop GUI's.
- Integrating database into your software (i.e., CSV files)
- How to hard code aggregation and association of the classes and use them in the coding process
- implementation of OOP concepts to the extent that all objects are connected to each other through object attribute for example having the chore list to know the assigned flat and assigned flatmate.

What still puzzles us?

- How to correctly integrate the software into the GUI?
- How to efficiently call the database when it becomes a lot bigger? (This wouldn't really apply for this software)

NOTES:

For future software engineers, please import the following "opencsv-5.6.jar"

The user is required to input the location of the project directory to be able to read the CSV files for example:

"D:\Bahaus univirsity\Software Engineering\Assignments\Second Submission\Flat-project-gradle-"