

Design approaches

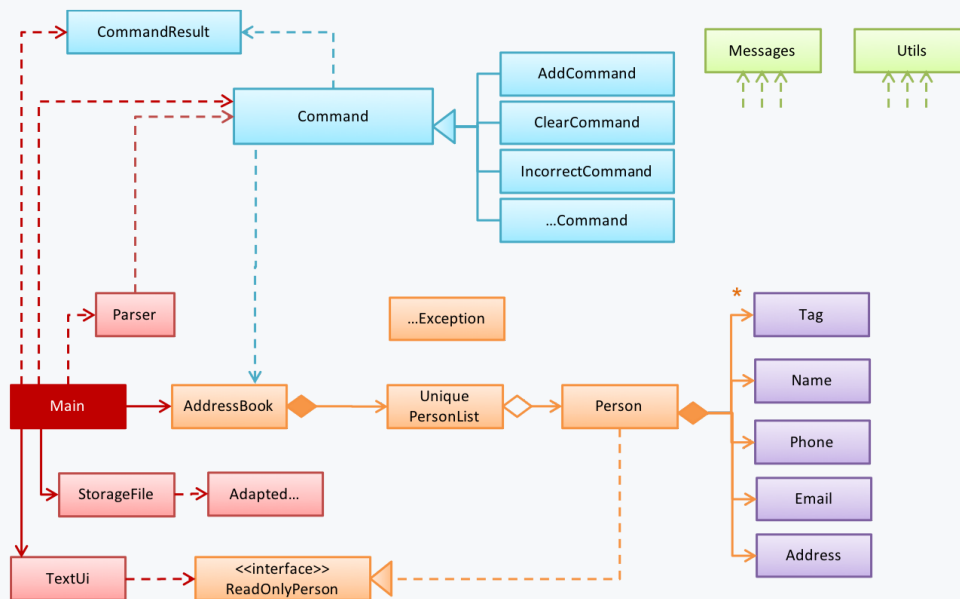
Multi-level design

What

☆☆☆ Can explain multi-level design

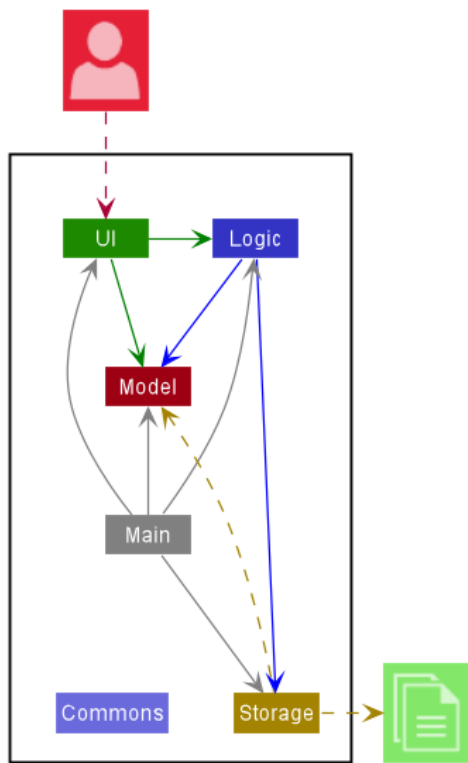
In a smaller system, the design of the entire system can be shown in one place.

This class diagram of [se-edu/addressbook-level2](#) depicts the design of the entire software.

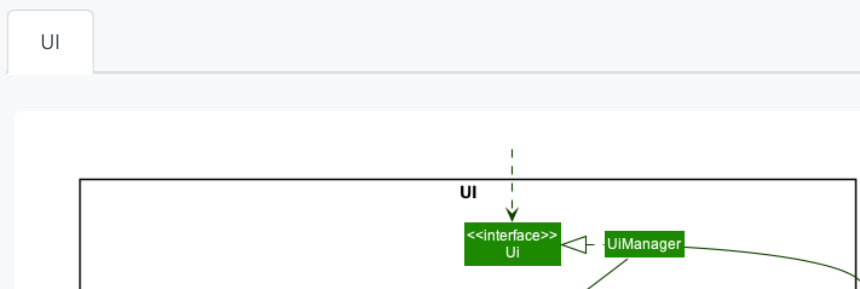


The design of bigger systems needs to be done/shown at multiple levels.

This architecture diagram of [se-edu/addressbook-level3](#) depicts the high-level design of the software.



Here are examples of lower level designs of some components of the same software:



▼ Top-down and bottom-up design

▼ What

★★★★☆ 🏆 Can explain top-down and bottom-up design

Multi-level design can be done in a top-down manner, bottom-up manner, or as a mix.

- Top-down: Design the high-level design first and flesh out the lower levels later. This is especially useful when designing big and novel systems where the high-level design needs to be stable before lower levels can be designed.
- Bottom-up: Design lower level components first and put them together to create the higher-level systems later. This is not usually scalable for bigger systems. One instance where this approach might work is when designing a variation of an existing system or re-purposing existing components to build a new system.
- Mix: Design the top levels using the top-down approach but switch to a bottom-up approach when designing the bottom levels.

🏠 Exercises



▼ Agile design

▼ What

★★★★☆ 🏆 Can explain agile design

Agile design can be contrasted with full upfront design in the following way:

Agile designs are emergent, they're not defined up front. Your overall system design will emerge over time, evolving to fulfill new requirements and take advantage of new technologies as appropriate. Although you will often do **some initial architectural modeling at the very beginning** of a project, this will be just enough to get your team going. This approach does not produce a fully documented set of models in place before you may begin coding. -- adapted from agilemodeling.com

🏠 Exercises

