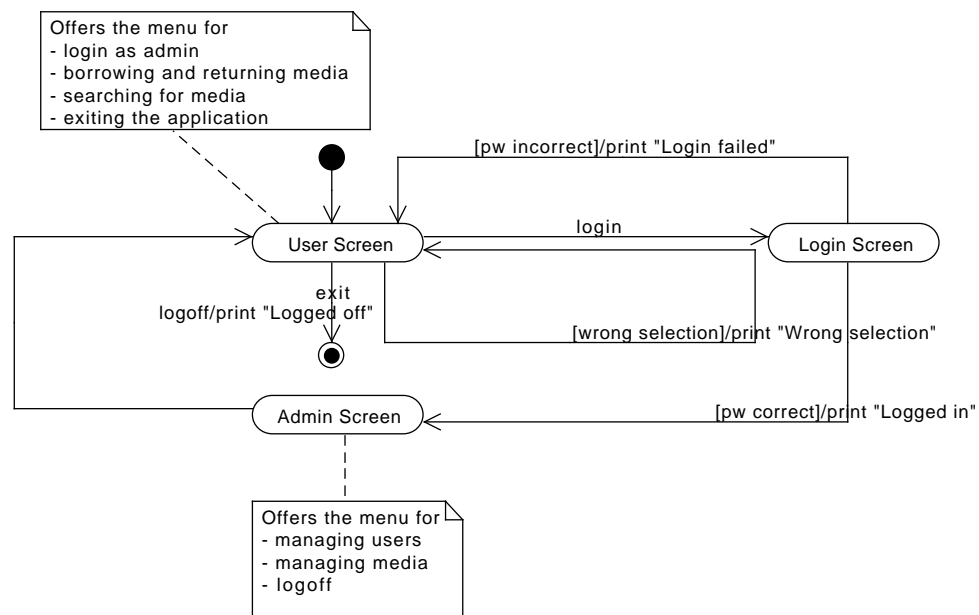


## 02161: Software Engineering 1

### Adding a command line user interface for the library application

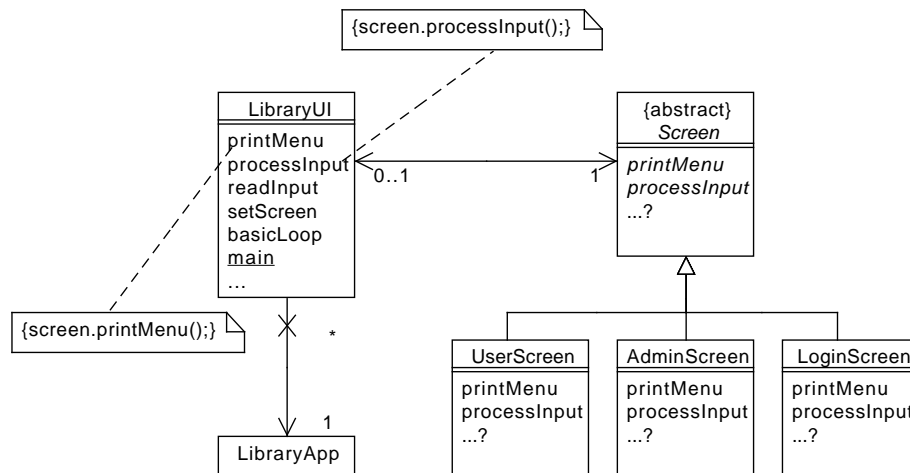
Up to now the application layer of the library application has been developed using test driven development. In the next step, a command line user interface should be designed and implemented. The interface should be designed using a state machine (cf. Web page of the course), where each state represents a screen and the transitions are the choices and inputs the user can make. Each input can result in some output and the transition to a new screen. The following is a state machine showing the UI for the user screen, the admin login functionality and the admin screen.



### 5.1 Implement the basic command line interface framework

- Download <http://www2.imm.dtu.dk/courses/02161/2015/files/library06.zip>. I have already provided the code from last week, but you can replace the code in `src/dtu/library/app` and `student_tests/dtu/library/app` with your classes and tests from the last exercises.
- Implement the UI tests by completing the `LibraryUI` class. Use the state pattern to implement the different screens depending on where one is in the implementation.

Note that the interactions should be similar to the state machine presented in the above. You can look at the course's Web page for some help



## 5.2 Extend the UI with the other functionality from the application layer

- Complete the state machine from above to include the remaining functionality of the application, implement tests for the user interface similar to the provided tests from the previous exercise, and implement the necessary screen classes. **Important: do these steps for each functionality separately.**
- List of functionalities
  - Add Book to library
  - Add CD to library
  - Register user
  - Search for media
  - Borrow media (no fines / no overdue media)
  - Return media
- You should implement as many of the previous functionalities as possible. If you have time left, you can complete the functionalities, e.g. borrow books when books are overdue, returning books that are overdue, paying fine, etc.
  - If you think that functionality is missing (and there is still a lot of functionality missing), you know how to implement them now:
    - 1) Create a test for the functionality for the application layer
    - 2) Implement the functionality in the application layer
    - 3) Design the user interaction based on a state machine
    - 4) Create a test for the user interface of the functionality

- 5) Implement the user interface for the functionality
- Again, don't forget to do each of the steps for each functionality separately, i.e., step 1 for the first functionality, then step 2, etc. and after that step 1 again for the next functionality