



eCOMMONS

Loyola University Chicago  
Loyola eCommons

---

Computer Science: Faculty Publications and  
Other Works

Faculty Publications and Other Works by  
Department

---

5-2016

## An Example of Atomic Requirements - Login Screen

William L. Honig

Loyola University Chicago, [whonig@luc.edu](mailto:whonig@luc.edu)

Follow this and additional works at: [https://ecommons.luc.edu/cs\\_facpubs](https://ecommons.luc.edu/cs_facpubs)



Part of the [Computer Engineering Commons](#), [Electrical and Computer Engineering Commons](#), and the [Software Engineering Commons](#)

---

### Recommended Citation

Honig, William L.. An Example of Atomic Requirements - Login Screen. , , , 2016. Retrieved from Loyola eCommons, Computer Science: Faculty Publications and Other Works,

This Working Paper is brought to you for free and open access by the Faculty Publications and Other Works by Department at Loyola eCommons. It has been accepted for inclusion in Computer Science: Faculty Publications and Other Works by an authorized administrator of Loyola eCommons. For more information, please contact [ecommons@luc.edu](mailto:ecommons@luc.edu).



This work is licensed under a [Creative Commons Attribution-NonCommercial-No Derivative Works 3.0 License](#).  
© 2016 William L. Honig.

## A Small Requirements Example

### Purpose

A simple example of making a requirements specification with numbered atomic requirements.

### System or Product Topic

Consider the familiar log in screen where a user begins access to a system. What atomic requirements can be generated?

### Working on Requirements

Example steps requirements team may go through in creating requirement(s) for this part of the system.

STEP 1 (initial concept):

Be sure the user is able to log in with password, log out, and reset password anytime.

Quality: low, likely better to separate some parts into individual requirements; terms such as password undefined; other problems.

STEP 2 (subdivide):

Be sure the user is able to log in with password anytime. (Other capabilities to be documented in separate atomic requirements).

Quality: low, lacks details; not clear or complete

STEP 3 (correct tone, more complete):

The system shall allow users to log in by providing a UserId and Password at the LogInScreen. The system shall check the UserId and Password provided to determine if the user is known to the system, in which case the user is allowed access to the MainMenu; otherwise, an error message is displayed and the system stays on the LogInScreen.

Quality: medium, provides more detail; includes multiple cases. Words shown in PascalCase are defined in the system Glossary.

STEP 4 (non functional requirements added; further error processing details):

The system shall allow users to log in by providing a UserId and Password at the LogInScreen.

- The system shall check the UserId and Password provided to determine if the user is known to the system, in which case the user is allowed access to the MainMenu; otherwise, an error message is displayed and the system stays on the LogInScreen.
- The set of currently known Userid 's and associated Password 's is stored in encrypted form inside the system; clear text of Userid and Password are never stored or saved inside the system. See Requirement 14 - User Administration for more details.
- If the user attempts to log in unsuccessfully using any Userid twice in 1 day, the user is warned that there is only one more opportunity to successfully log in before the account will be locked.

After the third failure to log in the Userid is locked. See Requirement 15 - Unlocking and Resetting User Identification.

- The system shall close the LoginScreen LoginTimeout seconds after it is displayed initially or after a failed log in attempt that resulted in the Userid being locked.

Quality: medium, beginning to be clear what the system needs the developer to do; defining what was meant by “anytime” in Step 1.

STEP 5 (adding other quality attributes, assigning identification number and brief name):

Requirement 1: Log in By User

Importance: Essential

The system shall allow users to log in by providing a UserId and Password at the LoginScreen.

- The system shall check the UserId and Password provided to determine if the user is known to the system, in which case the user is allowed access to the MainMenu; otherwise, an error message is displayed and the system stays on the LoginScreen.
- The set of currently known UserId 's and associated Password 's is stored in encrypted form inside the system; clear text of UserId and Password are never stored or saved inside the system. See Requirement 14 - User Administration for more details.
- If the user attempts to log in unsuccessfully using any UserId twice in 1 day, the user is warned that there is only one more opportunity to successfully log in before the account will be locked. After the third failure to log in the Userid is locked. See Requirement 15 - Unlocking and Resetting User Identification.
- The system shall close the LoginScreen LoginTimeout seconds after it is displayed initially or after a failed log in attempt that resulted in the Userid being locked.

Quality: high, still self contained or atomic with references to related requirements.

## Partial Glossary

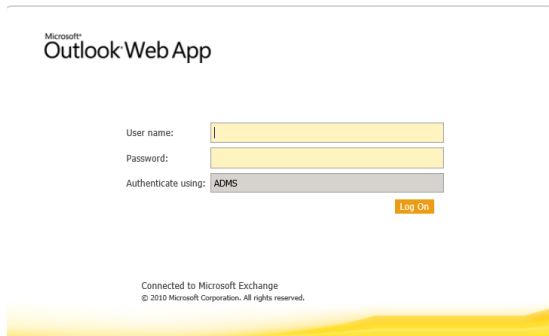
Include in the glossary any terms that will be used in describing the system especially if used in more than one requirement. Terms are always subject to misinterpretation unless carefully and precisely defined.

Work on the glossary proceeds at the same time as work on individual atomic requirements.

UserId: 6 to 10 characters at least one of which must be a number; characters one of 26 upper and lower case letters; numbers one of 0 to 9; upper and lower case characters are not distinguished (e.g. BOBBY12, bobby12, and bOBbY12 are the same id).

Password: 6 to 10 characters at least one of which must be a number; upper and lower case characters are distinct.

LoginScreen: ideal to do an actual GUI mock up with fields, labels, colors all shown. For example,



## Questions, Thoughts, ...

1. What does a “day” mean? Any 24 hour period? A clock day in some time zone?
2. What else can you think of that would make an improvement to Log in By User? Possibly more clear specification of what the user sees, for example when they type their user id and password? Other things?
3. Review the glossary carefully and suggest improvements. Should blanks be allowed in Userid and Password? What about that number 26? What character set (or sets) should be allowed?
4. Now we need to do the other two atomic requirements: User Log Out; User Password Reset. Do these suggest more details are needed Requirement 1 or is it now complete? Can Requirement 1 stand alone (with just the Glossary to support it)?

ExampleRequirements  
v1.0 (original) March 2016  
Dr. W. Honig