

A summary of the User interface specification standards – SE.QA.04:

- describes the format of and info to be supplied to the user
- describes components that make up the UI and how they are specified and delivered
- the objective is to aid the production of a UI specification which covers all of the requirements of the system it also specifies what the user can see and do throughout the use of the system
- the specification has 2 parts:
  - A use case Document this describes the main uses of the system, so basic layout, information content of this document must conform to the general documentation standards which specifies there must be a introductory section and reference section.
  - It contains the following specifications
    - Typical users there may be different types of users, e.g user, tester, developer ect
    - Use cases for each type of user there will be a set of tasks that they wish to carry out with the software all tasks should be listed and for each task write a description of how it is carried out and how the user can initiate the task and interact with it
    - Error conditions, it would be tedious to show all of the error conditions in the presentation but think about them, once done we need to plan how the UI will look to the user then for each use case figure out what could go wrong and list what is done about it
  - An interactive online presentation that shows the user experience for each use case. This should describe what is happening at each point in the interaction and show its results on the screen. This presentation does not need to try meet GDS but should have a version number and be stored in the repository
- Identify each possible type of user, writing about General user class, Specific types of user, Describe what different typical users want to do, Provide background detail for a fuller picture of importance,
- Example: Web site for parents of new babies
  - General classes's:
    - Contributors of articles
    - Readers of news
  - Sub-classes:
    - Contributors with extra responsibilities (e.g. able to delete articles)
    - Typical contributor, describe what they want to do for example:
      - Jennie, mother of two, writes about experiences as mother of Downs syndrome child, Submits articles through online form, preserving structure but losing formatting Adds formatting in online html editor, checks spelling, grammar, word limit before submission
      - Some applications (e.g. games) may only have one type of user
- Importance: Identify needs of all types of user
- Use cases
  - Document tasks for each type of user
  - Use cases for contributors:
    - Logging in (UC-1.1)
    - Submitting an article (UC-1.2)

- Editing an article
  - Deleting an article
  - Changing profile
- Use cases for editors:
  - Logging in (UC-1.1)
  - Adding new contributors
  - Deleting contributors
  - Featuring articles on the website
  - Plus all contributor use case
- Use cases for readers:
  - Selecting and reading an article
  - Marking articles as favorites
  - Finding a favorite article to read again
  - Give each use case a reference number for unambiguous reference
- Clear description of what happens with each use case
  - User initiation
  - User view
  - Completion
- Descriptions not tied to user interface to allow for changes
- Example: UC-1.2 Submitting an article
  - Log in with contributor ID (UC-1.1)
  - Press submit button
  - Fill in article title and subject, system records author name and date
  - Use HTML editor to fill in contents, preview, spell check available
  - Press Publish button to make article available
  - Receive confirmation, return to main contributor screen
- Error Conditions
  - Error conditions should be discussed and documented, including possible errors that can happen when the system is running and what can be done about them.
  - Each error condition should have an identifier (e.g., EC-1.1), a title and a description of how it is handled by the user interface, such as an error warning message displayed.
- Online presentation
  - The online presentation should be a mock-up of the application, giving a visual representation of each significant step in each use case, along with brief notations of what is happening.
  - The default way to create the online presentation is to use PowerPoint, but other specialized tools can be used as long as they deliver stand-alone deliverables that can be added to the GitLab repository.
  - The mock-ups should look as believable as possible, with buttons, etc. shown at the size they would be on the actual site, and content looking as much like the real application as possible.