**Mobile Application Self-Certification Initiative Training Materials**

**Veterans Health Administration**

**Human Factors Engineering**

March. 25, 2016

## Table of Contents

[Table of Contents 2](#_Toc446416837)

[Version History 3](#_Toc446416838)

[Training Materials Components 4](#_Toc446416839)

[Background 4](#_Toc446416840)

[Compliance Overview 4](#_Toc446416841)

[10 Heuristics 5](#_Toc446416842)

[Severity Ranking Scale 7](#_Toc446416843)

[An Overview of the Self-Certification Process 7](#_Toc446416844)

[Mobile User Interface Design Certification 7](#_Toc446416845)

[Heuristic Evaluation 7](#_Toc446416846)

[Peer Review 8](#_Toc446416847)

[Download to Repository 8](#_Toc446416848)

[Process Flow Chart 9](#_Toc446416849)

[Mobile User Interface Design Certification 10](#_Toc446416850)

[Category Questions 10](#_Toc446416851)

[“Test Yourself” – UI Cert Instructions 12](#_Toc446416852)

[Heuristics 12](#_Toc446416853)

[Heuristic A: Visibility of system status 12](#_Toc446416854)

[Heuristic B: Match between system and the real world 13](#_Toc446416855)

[Heuristic C: User control and freedom 13](#_Toc446416856)

[Heuristic D: Consistency and standards 14](#_Toc446416857)

[Heuristic E: Error prevention 16](#_Toc446416858)

[Heuristic F: Recognition rather than recall 17](#_Toc446416859)

[Heuristic G: Flexibility and efficiency of use 18](#_Toc446416860)

[Heuristic H: Aesthetic and minimalist design 19](#_Toc446416861)

[Heuristic I: Help users recognize, diagnose, and recover from errors 19](#_Toc446416862)

[Heuristic J: Help and documentation 20](#_Toc446416863)

[“Test Yourself” – Heuristic Evaluation Instructions 21](#_Toc446416864)

[“Test Yourself” Comparison 23](#_Toc446416865)

[Review / Peer Review and Posting 23](#_Toc446416866)

[Appendix A: Severity Ranking Scale 24](#_Toc446416867)

[Appendix B: Heuristic Evaluation Job Aid 26](#_Toc446416868)

[Appendix C: HFE Test Certification Wireframes 28](#_Toc446416869)

[Appendix D: UI Test Certification Comparable Findings 32](#_Toc446416870)

[Appendix E: HE Test Certification Comparable Findings 32](#_Toc446416871)

[Appendix F: Help for Project Teams 32](#_Toc446416872)

# Version History

|  |  |  |
| --- | --- | --- |
| Version | Date | Comments |
| V 0.1 | 01/15/2016 | Initial draft |
| V 0.2 | 01/20/2016 | Initial review and standardization of draft materials |
| V 0.3 | 01/28/2016 | Compilation of all project materials & incorporation of VA management comments for training materials |
| V 0.4 | 01/29/2016 | Incorporation of review feedback |
| V 0.5 | 01/29/2016 | Incorporation of white glove review feedback |
| V 0.6 | 02/29/2016 | Second round of feedback received |
| V 0.7 | 03/02/2016 | Incorporation of second round of feedback |
| V 0.8 | 03/22/2016 | Incorporation of third round of feedback |

# Training Materials Components

## Background

Development of mobile applications for use by Veterans currently follows a strict compliance process within Veterans Affairs. This begins with Validation and Verification (V&V) and then proceeds onto a number of compliance bodies including 508 Accessibility, Patient Safety, and Human Factors Engineering (HFE). In an effort to simplify the process from the developer’s perspective, HFE has created a process to serve as a guide towards self-certification. Materials in this document will teach vendor developers or project managers the process that HFE uses during compliance reviews of a mobile application so that they may apply those principles during development.

Upon completing the work outlined in this document vendors will be able to self-certify that an app developed within their organization adheres to the same standards that HFE tests against during the certification process. The actual developers of an app should **not** do the self-certification of their own app. The self-certification process is ideally performed by team members of the developer. This could include project managers, development coordinators, or developers of *other* apps. This document will begin with a brief overview of all aspects of the compliance process and then go more in depth into each aspect of the self-certification.

# Compliance Overview

An HFE compliance review of an application is composed of two primary parts: the Mobile User Interface (UI) Design Certification (UI Cert) and the Heuristic Evaluation (HE). The HE is performed through reviewing an app while being mindful of 10 industry standards for heuristics, and assigning a severity rating to issues that violate the said standards. The UI Cert is performed by following a strict criteria list which ensures the app meets key design standards and documenting any deviations the app has against the listed criteria with an appropriate severity rating.

The review is always performed on one of the intended device types for the app and upon completion of a review; it is passed on to a co-worker for a peer review. The peer review entails the same process as the initial review, but is performed on a different device from the initial review ( e.g. if the initial review was performed on an Apple device, the second review is performed on an Android device, or if the initial review was done on a phone, then second review is on a tablet). This ensures a different perspective of the app in addition to uncovering issues that may only surface on one operating system (OS) or device type. Lastly, the HE should be performed from the perspective of applicable personas. A persona is a specific individual who represents the needs and expectations of a larger group (but is not a real person). To simplify the effort for developers, only one persona will be used per app.

## 10 Heuristics

The 10 heuristics that an app is reviewed against are an industry standard developed by Jakob Nielsen in 1995[[1]](#footnote-1) for user interface design. These are considered 10 core principles that an application, mobile or otherwise, should follow to maintain a high level of usability and user satisfaction. A listing of the 10 heuristics is below, but these will be reviewed more in depth later in this document.

|  |  |
| --- | --- |
| **Heuristic Code** | **Heuristic(s)** |
| A | **Visibility of system status** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time. |
| B | **Match between system and the real world** The system should speak the users' language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order. |
| C | **User control and freedom** Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo. |
| D | **Consistency and standards** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. |
| E | **Error prevention** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action. |
| F | **Recognition rather than recall** Minimize the users’ memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. |
| G | **Flexibility and efficiency of use** Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. |
| H | **Aesthetic and minimalist design** Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility |
| I | **Help users recognize, diagnose, and recover from errors** Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. |
| J | **Help and documentation** Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large. |

Table 1: 10 Usability Heuristics for User Interface Design

## Severity Ranking Scale

The severity ranking scale used by HFE is an adaptation of the scale developed by Jakob Nielsen in 1995[[2]](#footnote-2) which evaluates the severity of usability problems against three primary factors:

* The impact of the problem if it occurs: Will it be easy or difficult for the users to overcome?
* The frequency with which the problem occurs: Is it common or rare?
* The persistence of the problem: Is it a one-time problem that users can overcome once they know about it or will users repeatedly be bothered by the problem?

The ranking scale produced by HFE was modified to be more applicable to the VHA, and it includes language changes and increased clarifications to reduce potential confusion. The same rankings are used for both HEs and UI Certs. Severity rankings are shown in the [Appendix A](#_Appendix_A:_Severity).

# An Overview of the Self-Certification Process

## Mobile User Interface Design Certification

The Mobile UI Design Cert is performed by following a checklist which evaluates the app in eight categories with 28 questions. The checklist is a combination of UI design references among VA/VHA, Apple, and Android. This was done to ensure that an app will meet internal VHA guidelines that have been set, while still meeting all applicable OS standards for mobile app design. Following this is the Findings section where the reviewer can document whether the app met the criteria and if not, at what severity the app deviated from it. Comments are added as to how the issue(s) could be remedied in addition to a screen capture to provide additional clarification. Many criteria listings also include guidance and examples help the reviewer identify violations and/or severity levels.

## Heuristic Evaluation

An HE is a usability inspection method performed to help identify usability problems in the user interface design. The person doing the review will examine the app while judging it against the 10 heuristics previously mentioned, and then look at every aspect of the application, testing all menus, links, forms, and entry fields to determine if everything works as it should. In a sense, the reviewer is testing the app with the intent to “break” it and find any flaws.

As the reviewer goes through the app and finds issues, they document what the specific issue is, provide an accompanying screenshot for further clarification, decide which heuristic(s) it violates, assign a severity rating, and recommend how the issue should be remedied. As mentioned before, the review is performed from the perspective of an applicable persona, as the intended user of the app would likely have a very different perception from the developer of an app.

## Peer Review

The peer review portion of certification is a very important step. As with any review, a second set of eyes is always better than one. A peer could be a project manager, another developer, or a coordinator on that app or another project. Upon finishing the HE and UI cert for an app, the initial reviewer passes it off to a peer to perform a second review completely independent of the first, while using a different applicable device type and/or OS. Issues that may be very prominent on an Apple tablet may never appear on an Android phone, which is an important distinction to document for the development team. Upon completion of the peer review process, the two reviewers need to agree on findings and severity levels. All applicable HE and UI violations are combined into one document which then constitutes a completed certification review.

## Download to Repository

When the review is completed and peer-reviewed, the developer uploads it to the [designated repository](https://vaww.portal2.va.gov/sites/humanfactors/SelfCertification%20Reviews/Forms/AllItems.aspx) site (which requires VA access). At this point, the self-certification is complete. After uploading the document it is recommended to keep a local copy as well to mark and track changes made to the application based on the findings.

# Process Flow Chart

Refer to this flow chart to get an idea of the process, as well as to ensure that you complete the full process, including posting the final document.

Flow Chart of the Self-Certification Process

Figure 1: Flow Chart of the Self-Certification Process

# Mobile User Interface Design Certification

Note: *The draft materials below will be provided to the development team that is doing the self-certification.*

The Mobile UI Design Cert is performed by following a checklist that evaluates the app in eight categories with 28 items. An example of the form used to complete the UI Cert is show below in Figure 1. The Excel workbook UI tab starts with the specific category being evaluated, the ID for the item in the category, and the criteria used to evaluate. Following this is the Findings section where you, the reviewer, document whether the app met the criteria and if not, the impact (severity) of the deviation. The Comments box is used to provide additional context regarding the finding, while the Screen Capture(s) provides additional clarification to the issue and specific location. The Guidance box provides applicable information to help you determine whether a criterion was met.

The image shows an example of the UI Cert workbook. The order of the various fields are as follows:
Category, ID, Criteria, Finding, Comments, Screen Capture(s), Guidance

Figure 2: The Mobile UI Design Cert Workbook Format (partial)

## Category Questions

The eight categories included in the UI Cert encompass the core aspects of an app that need to be evaluated during a UI certification. More guidance is also provided in the workbook, which present a series of questions to be answered. These categories (and questions) include:

1. Consistency – Terminology, standard navigation, lay-out, content presentation, load times, gestures, VA branding, icons:
   * Is terminology used consistently throughout the application?
   * Do application screens include standard navigation, adhering to the convention for either native or browser-based apps?
   * Are layout and content presentation consistent throughout the application?
   * Do all methods of scrolling and screen movement look and behave consistently throughout the application?
   * Does the application provide immediate (.25s) response (visual, audio, haptic feedback) indicating user input?
   * Does the application provide a processing indicator (e.g. a twirling wheel) when tasks take between 1-10 seconds?
   * Does the application provide an indication of remaining processing time for tasks expected to take 10 seconds or longer (e.g. saving information)?
   * When launching an application, does a splash screen convey it is a VA-sanctioned application?
   * Do all core gestures provide standard functionality, following the conventions for native or browser-based apps?
   * Does the application use standard icons that conform to VA branding and conventions for native or browser-based apps?
2. Device Orientation – portrait vs. landscape choices:
   * Does the application provide a user experience consistent with the referenced device or OS guidelines for single or multiple screen orientations (portrait and/or landscape)?
3. Errors – Issues described, issue recovery:
   * Are application messages (e.g. error, warning, informational, alerts) written simply and plainly?
   * Do error messages plainly and precisely describe the problem and how to recover?
4. Frequent Interactions – Clear navigation, all functionality available:
   * Does every screen have a clear path to the next step in the activity or, when appropriate, access to other relevant activities?
   * If the app opens in 'standalone' [iOS] or 'full screen' [Android] mode [when placed on home screen], does it allow user to perform all of the app's functions?
5. Modal Tasks – Task abandonment, button displays, tasks identified:
   * When a user task presents a modal screen, is the user given the choice to complete the task or abandon the task (no changes are effected)?
   * When a user task presents an action sheet (a set of options for selection by the user), can all buttons be displayed without requiring the user to scroll?
   * When a user task presents a modal screen, does the title identify the task that required the screen to exist?
6. Readability – External links properly denoted, truncation avoided, text size, scrolling, space “used” by controls:
   * Do links correctly indicate the destination sites to which they navigate?
   * Are all menu items short enough to avoid truncation when displayed on the application’s target device?
   * Does the application icon, appearing on the device, have the entire application name visible (no ellipses) and in the same font size as any other application name?
   * Is all text large enough to be easily read?
   * Is it evident that the user needs to scroll to see all content on a page?
   * Do controls take 20% or less of screen space with the application information consuming the remainder of the screen space?
7. Sound – Volume adjustment:
   * When sound is inherent to application functioning (e.g. video or audio clip), can the user adjust volume levels based on their preference?
8. User Input – Proper size on tappable elements, active entry field indicators, menu design:
   * Are all tappable elements at least 10 mm on one side (either 10 mm diameter, or 10 mm vertical or horizontal)?
   * Do active text entry fields have an input indicator?
   * Does the app avoid more than one row of (menu) buttons at the top and bottom of the device screen [per OS guidelines]?

# “Test Yourself” – UI Cert Instructions

You will now perform a UI Cert on your own using the provided [HFE Test Certification Wireframes](#_Appendix_C_–). Please go through the wireframes to the best of your ability, answering the questions while judging it by the criteria in the provided Excel file. Please not that due to the nature of wireframes, some questions may be Not Applicable. You should also make comments relating to the remediation of any issues you find. For this test you do **not** have to take screen captures of the issues you find. After you finish, the [Appendix D](#_Appendix_D:_UI) of this guide will go through the correct findings with you so that you can assess how well you did.

# Heuristics

To acquaint you better with the heuristics, examples of each are below with actual findings, taken from HFE-reviewed apps.

References to HFE personas (Dan, Joyce, Byron, etc.) are shown. These personas were rigorously developed to represent the many clinician and Veteran users of VHA apps. For this test only, you do not need to reference personas.

## Heuristic A: Visibility of system status

**The system should always keep users informed about what is going on, through appropriate feedback within a reasonable time.**

**Example Finding:** There is no "Send" button or "x" to close out the tip entry screen on the iPad, so this effectively locks users into the area and forces them to perform a hard restart of the app. On the iPhone, a light blue "Cancel" button appears but because it is on a dark blue background, there is insufficient contrast for users who may have vision problems. **Serious ranking**

**Example Recommendation:** Add the "Send" button and a way to exit the tip window on the iPad. For the iPhone version, improve the contrast for buttons on the dark blue background; this applies to several buttons on several screens.

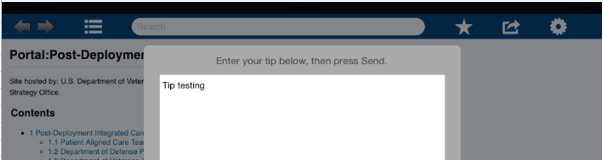
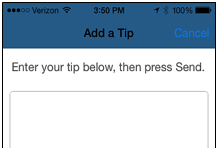
 

Figure 3: Difficult to close out of iPad screen, but Cancel provided on the iPhone version

## Heuristic B: Match between system and the real world

**The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.**

**Example Finding:** The ‘hamburger’ icon is used to toggle between graph and table views. This will likely confuse Joyce, who may expect the hamburger to represent a menu due to past experience with this icon. **Moderate ranking**

**Example Recommendation:** Look at using another icon to depict the table view.

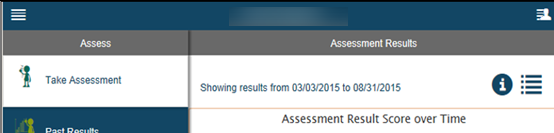


Figure 4: Use of multiple hamburger-like icons is confusing

## Heuristic C: User control and freedom

**Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.**

**Example Finding:** When accessing the "User Menu" at the top right, Joyce is limited to just the menu choices from that menu. She won't be able to swipe or move to the other menu, except by tapping on "User Menu", which does not appear to be a button. **Moderate**

**Example Recommendation:** Consider another way for Joyce to navigate back to the main menu for the app.

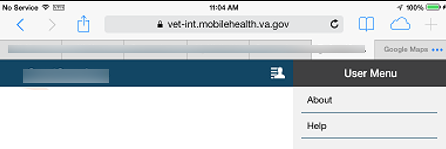


Figure 5: Confusing navigation

## Heuristic D: Consistency and standards

**Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.**

**1 Example Finding:** Under the "App Options" menu, the Favorites icon is an unfilled star, but items that have been favorited have a gold star. This could lead to confusion for Joyce as the unfilled star represents items that aren't currently a favorite. **Minor**

**Example Recommendation:** Make the Favorites icon under "App Options" gold to be consistent with items that have been marked as a Favorite.

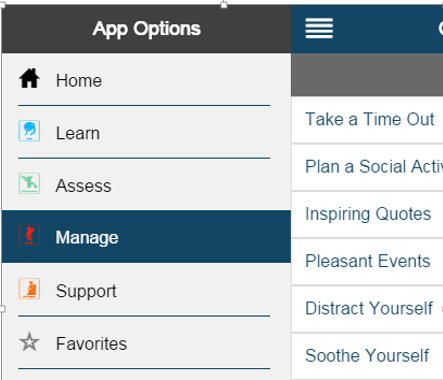
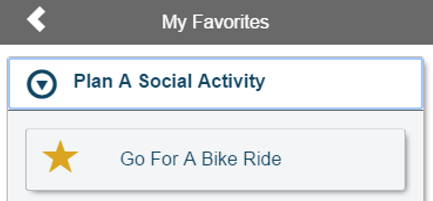
 

Figure 6: Inconsistent use of stars

**2 Example Finding:** The term "Tools" is used on the "Relax Yourself" screen, but the tools involved are really relaxation videos, which could confuse Joyce. **Minor**

**Example Recommendation:** Consider using the term "Videos" to describe them.

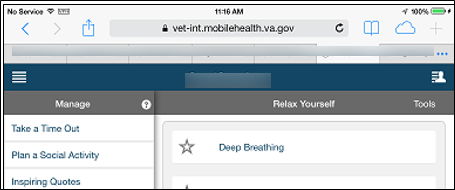


Figure 7: Confusing, inconsistent use of the term "Tools"

**3 Example Finding**: "Contact Support" doesn’t automatically hyphenate phone number entries on the iPad, but works on the iPhone. **Minor**

**Example Recommendation:** Auto-hyphenate a phone number as the user enters it. Provide consistency between devices.

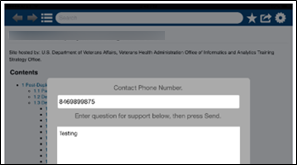
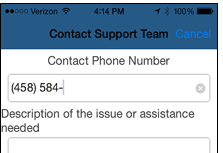
 

Figure 8: Inconsistent auto-hyphenate functionality

**4 Example Finding:** On the "Post-9/11" screen, the text "stipend" is underlined, and when it is tapped, a tooltip comes up with a definition of the stipend. Using web conventions, underlined text generally indicates a link to another source, not a tooltip. There is also no way to close the tooltip except by clicking back on the text. **Minor/Moderate**

**Example Recommendation:** It is a plus for the Veteran like Byron to be able to access definitions of terms that he needs, but use common conventions to indicate links and definitions (such as underlining for links and an “i” icon for information). Add an "x" to enable the tooltip to be closed.

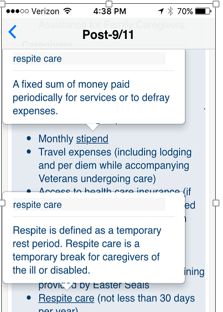


Figure 9: Tooltips accessed by clicking on underlined text, a non-standard usage

## Heuristic E: Error prevention

**Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.**

**Example Finding:** When viewing favorites, selecting the star icon itself will remove the activity as a favorite without any prompt. This action could cause Joyce to accidently lose one of her favorites when attempting to select it for use. **Moderate**

**Example Recommendation:** Add in a prompt to confirm removal of favorites so as to avoid accidental removal.

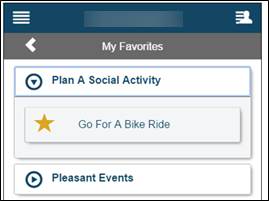


Figure 10: Potential for error by tapping the star

## Heuristic F: Recognition rather than recall

**Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.**

**1 Example Finding:** It isn't possible to see the current activity box and the control buttons at the same time in the phone form factor. Meghan and Byron are most likely to use the activity pacing while carrying a smaller device. They should be able to see their current interval and control the activity player. **Moderate**

**Example Recommendation:** Consider affixing the player to the bottom of the phone form factor and allowing the rest of the page, including the intervals to scroll behind it until the bottom of the page, where the user can then scroll past the controls to the bottom.

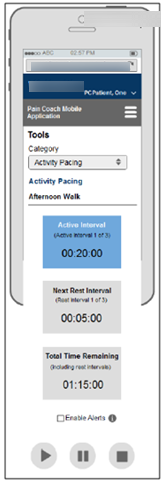


Figure 11: Up-and-down scrolling necessary to use screen

**2 Example Finding:** The lack of labels on the many different graph axes could prevent the user from accurately interpreting the data. All of the personas who might use the app may have trouble remembering and interpreting the values and date ranges conveyed in the graphs, an otherwise powerful method of demonstrating trends. **Moderate**

**Example Recommendation:** Add x and y axis labels to provide exact context for the highs and lows and date range.

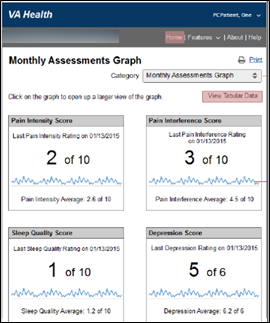


Figure 12: Memory load to interpret many graphics may tax users

## Heuristic G: Flexibility and efficiency of use

**Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.**

**Example Finding**: All the page headings (e.g., the "About", "Support", "Reason", etc.) function as a "back" button. These actually take the user back to the home menu. For advanced users, a fast way to go back would be a plus, but an intuitive, standard way needs to be provided. **Minor/Not Applicable**

**Example Recommendation:** Separate page navigation from labels and make button actions explicit. Since device controls are recommended for app navigation any "back" functionality should be limited to sub-content or to the main menu if explicitly labeled.

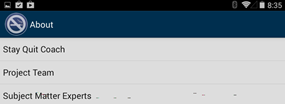
****

Figure 13: Non-standard back button accelerator usage

## Heuristic H: Aesthetic and minimalist design

**Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.**

**Example Finding:** The "Relax Yourself" videos are well done, but the video screen is very small and the text is too small to view comfortably, especially for an older caregiver like Joyce. Additionally, the large text stating “listen to the audio and follow along” may be unnecessary if the embedded video is obvious, since Joyce may readily recognize it. **Minor**

**Example Recommendation:** If feasible, fill up more of the screen with the video image so that Joyce can immerse herself more fully in the relaxing scene and also read the accompanying text.

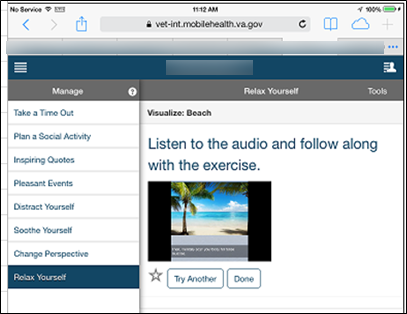


Figure 14: The video picture and text are too small and the large instruction text may be unnecessary

## Heuristic I: Help users recognize, diagnose, and recover from errors

**Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.**

**Example Finding:** The reviewer received a 403 error message when logging in to the app. Receiving such a message may cause Dan or Joyce to give up and never access the app in the future. The message does not tell the user how to recover from the error condition. **Moderate**

**Example Recommendation:** Discover the reason for the error and tell the user how to fix it.

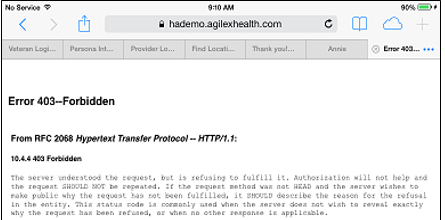


Figure 15: The error message does not give the user a way to fix the issue

## Heuristic J: Help and documentation

**Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.**

**Example Finding:** The learning curve for the application is very steep. There's no one element that makes it hard to use, it is just a lot to learn, which in turn can frustrate users. The lengthy 64-page User Manual is useful, but most users may not have the time to read it and there is no contextual help. The Help Desk number and link are also missing. **Moderate**

**Example Recommendation:** Look into producing a shorter guide and contextual help that can easily be accessed by users.

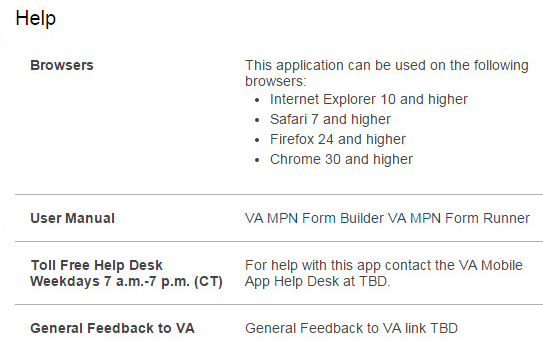


Figure 16: Help information for users is missing

# “Test Yourself” – Heuristic Evaluation Instructions

1. You will now test your understanding of the heuristics using the [HFE Test Certification Wireframes](#_Appendix_C_–). Refer to the [process flow chart](#_Process_Flow_Chart) if you wish.
2. Go through the heuristics and the provided wireframes, and fully document at least five or six heuristic violations that you find in the wireframes. Follow the steps below:
3. Open the Self-Certification template (link), and document your findings in the Heuristics tab and the Finding column.

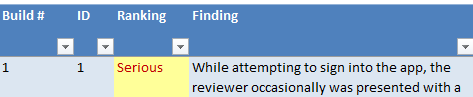


Figure 17: Self-Cert Template, showing Finding column (partial)

1. Add the heuristic code in the Heuristic Code column for each one. If more than one heuristic applies to the finding, include as many as are applicable.

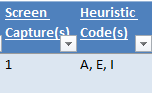


Figure 18: Heuristic code column

1. Then, take a screen shot of the finding and put it in Screen Captures tab. Each screen shot needs to be placed in a separate area so that the number of screens can be tabulated. You may add red arrows, circles or comments to highlight the area of the screen with the finding.

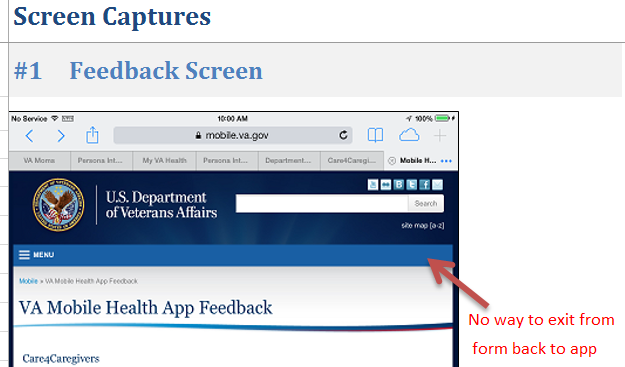


Figure 19: Screen Captures tab

1. Note the number of the screen capture and put it in the Screen Capture column on the Heuristics tab. If you wish, use a red arrow or circle to highlight the finding.

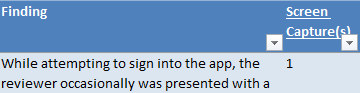


Figure 20: Screen Captures column

1. Now, write your recommendation(s) to fix the problem, and put it in the Recommendations column.

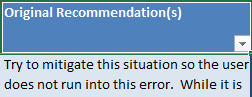


Figure 21: Recommendations column

1. Finally, select a ranking from the Rankings dropdown, using the guidance and examples from the [HFE Heuristic Rankings Matrix](file:///C:\Users\William%20Plew\Downloads\Self-Cert%20Training%20Materials%20Jane%20stuff.docx#_Appendix_A_–).

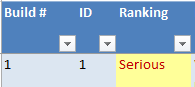


Figure 21: Rankings column

1. The template will automatically calculate and graph the findings by severity on the Report Summary tab. At the end of the process, a graphic with metrics will appear on that tab. Check to make sure that the total is at least five.

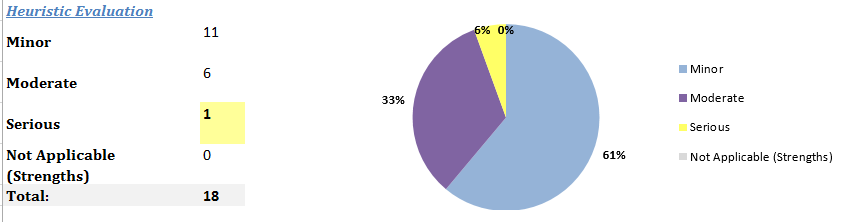


Figure 22: Report Severity tab, once rankings are done

## “Test Yourself” Comparison

Now it’s time to check your results. Refer to [Appendix E](#_Appendix_E:_HE) to see what an HFE expert found when doing a heuristic review. Look at the differences between what you found and what the HFE expert documented. Note that once you’ve done this a few times, it will get easier and seem natural, and you may find yourself observing heuristic violations even without reference to the chart! Also, a [job aid](#_Appendix_B:_Heuristic) has been prepared that you can use to quickly find a heuristic.

# Review / Peer Review and Posting

After finishing your UI and HE reviews, have another member of your team (a project manager, another developer, a coordinator, etc.) also perform an independent UI and HE review. As previously noted, they should do their review on a different device or OS, if possible.

The easiest process involves sending them your Excel review document and then have them check to ensure that they also find the same issues that you found, and then add additional issues that they find. Then, get together and discuss and agree upon all of the findings and rankings for each. Finalize the self-certification document and post it to the designated [HFE repository](https://vaww.portal2.va.gov/sites/humanfactors/SelfCertification%20Reviews/Forms/AllItems.aspx) (this link requires VA access).

## 

## Appendix A: Severity Ranking Scale

The severity ranking scale used by HFE is an adaptation of the scale developed by Jakob Nielsen in 1995[[3]](#footnote-3) which evaluates the severity of usability problems against the three primary factors of frequency, impact, and persistence.

|  |  |  |  |
| --- | --- | --- | --- |
| **Ranking** | **Definition** | **Recommended Priority for Resolution** | **Examples** |
| **Minor** | **One of more** of the following: □ Causes user hesitation, confusion, or slight irritation. □ Impedes task completion or decreases efficiency but does not cause task failure. □ Presents small likelihood that the credibility of the VA HIT product will be diminished. | Consider resolving this issue. | Use of “Click here for more” to take user to an external link. |
| **Moderate** | **One or more** of the following: □ Causes occasional task failure after which recovery is possible. □ Causes user delays and/or moderate dissatisfaction, but some users are able to recover in order to complete the task. □ Expected to negatively impact use, possibly leading to dissatisfaction at a level that users might opt to discontinue use.  □ May diminish the credibility and/or reputation of the VA product. | Give high priority to resolving this issue. | Inconsistent access to app navigation (e.g., menu button alternates between the right and left side, depending on page). |
| **Serious** | **All** of the following: □ Causes frequent task failure or occasional task failure from which recovery is not possible. □ Causes extreme user irritation and/or task abandonment.  □ Likely to diminish the credibility or reputation of the VA product. **Or:** □ Causes system/sub-system failure (i.e., produces system error or “crash”) | Give highest priority to resolving this issue prior to further product testing or release.  ***HFE recommends resolution or mitigation for serious usability issues before deploying products.*** | Blank page in app. Broken [external] web link (e.g., link has changed). Inaccessible web link (e.g., link is behind firewall, but app user is not). Use of language that is not easily comprehended by end users. |
| **Not Applicable** | In HE: Strengths or Unsolicited Suggestions Any findings related to strengths in the system (or unsolicited suggestions for improvement, which are not related to a usability weakness).  In UI Cert: Criteria not applied (doesn’t exist within the app). | Optional. | UI Cert:  If app does not have sound features, criteria S-01 is not applicable If app is 'read only,' criteria I-02 is not applicable If app does not support processing tasks, criteria C-07 is not applicable |
| **Not Evaluated** | In UI Cert: Reviewer did not come across (observe) criteria, so cannot say if criteria is met or not. This can be because function is not working or reviewer could not create the circumstances to support evaluation. |  | UI Cert: If reviewer does not come across error messages, criteria E-02 is not evaluated (because a different set of key presses or network speed might allow for it to be evaluated at another time) If reviewer does not come across pages that take 1-10 seconds to load, criteria C-06 is not evaluated (because a different set of device + network speed might allow for it to be evaluated at another time) |

## Appendix B: Heuristic Evaluation Job Aid

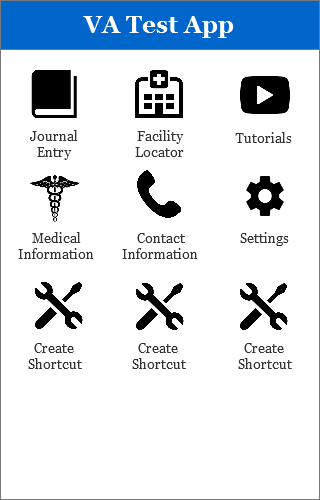
This job aid can be printed or saved to enable a quick lookup of the heuristics and examples.

|  |  |  |
| --- | --- | --- |
| Heuristic Code | Heuristic | Explanation & Examples |
| A | **Visibility of system status** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time. | Within the app, users should always know where they are within the app’s architecture as well as the status of any transaction they are doing within the app. For example, if the user is downloading something, a status indicator should be provided. |
| B | **Match between system and the real world** The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order. | This is one of the most commonly cited heuristics. Essentially, the app should use terms and graphics commonly known to the targeted user. For example, using a stop sign or the color red to prevent an action might be appropriate, whereas using a green sign would not. Using an unfamiliar, offensive or technical term in an error message are examples of not matching the user’s world. |
| C | **User control and freedom** Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo. | App users always need to be able to exit from screens within the app as well as modal dialog boxes that may be presented. Such boxes need to have a cancelation option so that the user can back out of it and return to the previous screen. |
| D | **Consistency and standards** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. | Consistency is a major aim when performing a heuristic evaluation. Terminology, graphics, navigation and layout, as well as general look-and-feel, need to be consistent throughout, using OS standards. |
| E | **Error prevention** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action. | Test out all possible conditions that could create errors and provide warnings or another way so that the user never encounters them. Use error messages only when necessary, by reducing the number of times that they are possible. Also see Heuristic I. |
| F | **Recognition rather than recall** Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. | Reduce the user’s cognitive load by allowing the user to easily refer to previous information and also by providing links to information directly when it is needed. Also see Heuristic J. |
| G | **Flexibility and efficiency of use** Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. | Although it’s unusual to see specific accelerators in apps, always use conventional OS gestures (swiping, tapping, etc.) as appropriate to speed up the use of the app. If possible, offer a brief tutorial for any unusual actions that the user needs to do, preferably at first logon or at the time the user needs the information. Allow the option to set up customized parameters if applicable. |
| H | **Aesthetic and minimalist design** Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. | Simplify the design to encourage the user to focus on the most important aspects of each screen of the app. Still, ensure that any less frequently used information is available as needed. Make sure that controls and buttons are large enough to see but don’t dominate the screen space. Use responsive design principles to ensure that users on all devices can use the app efficiently. |
| I | **Help users recognize, diagnose, and recover from errors** Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. | Make sure that error messages are primarily non-technical and show the problem exactly, as well as how to resolve it. As applicable, include resources (such as a help desk) that could provide further assistance. |
| J | **Help and documentation** Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large. | If the app is complex or the target audience may be unfamiliar with the information provided in the app, include help resources for the app. Although general tutorials and references are helpful, focus on providing help at the time the user needs them. |

## Appendix C: HFE Test Certification Wireframes

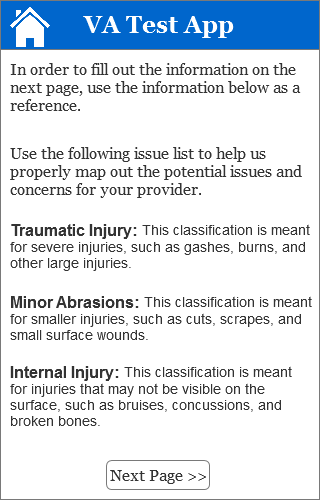
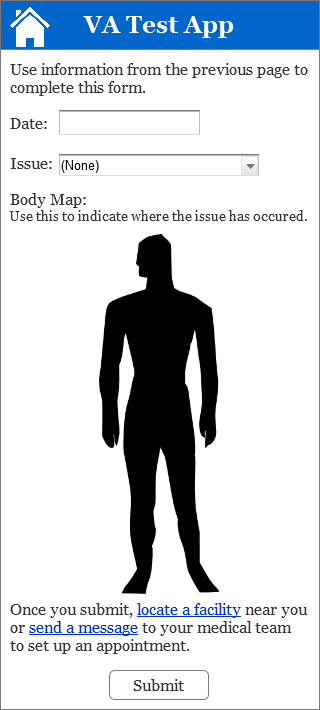
These wireframes should be used for your UI Certification Test and HE Test. For the purpose of testing, consider the wireframes to be an app developed for Veteran use, and test it from the perspective of a chosen Veteran persona. A brief overview of each screen is provided along with a “flow” to provide additional context.

Screen 1: Home Page of the new VA Test App for Veterans. From here the user can navigate to various widgets.



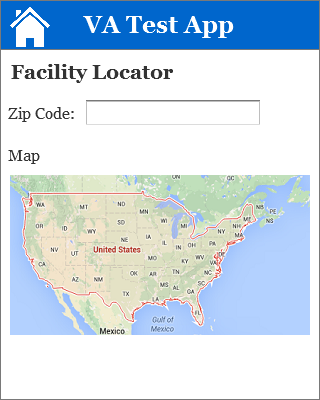
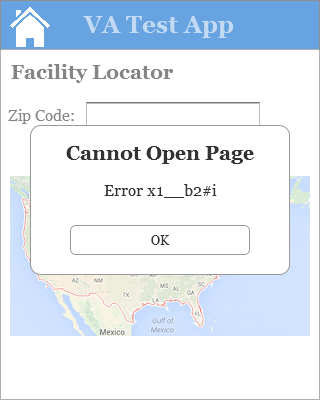
Screen 2 is the Journal Entry screen accessed from the Home Page. Here the user is given a brief instructional overview of how to enter issues.

Screen 3 is the follow up page to the Journal Entry screen where the user can indicate their various injuries.

**** 

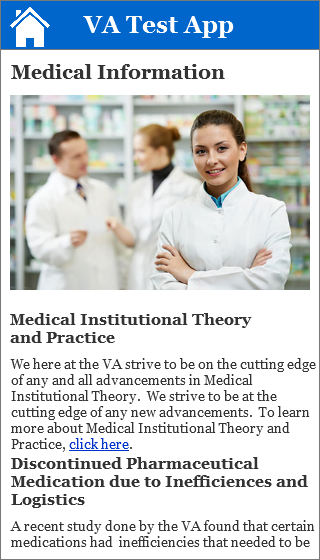
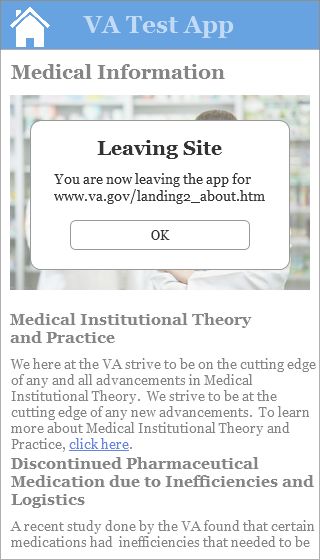
Screen 4 is the Facility Locator accessed from the Home Page or link at the bottom of Screen 3.

Screen 5 displays an error that occurred when the user attempted to enter a zip code on Screen 4.

**** 

Screen 6 is the Medical Information screen accessed from the Home Page.

Screen 7 is the message that appears when the user selects “click here” on screen 6.

** **

## Appendix D: UI Test Certification Comparable Findings

As a number of findings are not-applicable with wireframes as opposed to an actual app, any items that were **NOT** flagged as findings have been hidden. Simply compare your Criteria ID number to those in the attached worksheet.



## Appendix E: HE Test Certification Comparable Findings



## Appendix F: Help for Project Teams

For questions about the process or criteria, contact the HFE team for assistance at [Vha10p2hfq@va.gov](mailto:Vha10p2hfq@va.gov).

1. https://www.nngroup.com/articles/ten-usability-heuristics/ [↑](#footnote-ref-1)
2. https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/ [↑](#footnote-ref-2)
3. https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/ [↑](#footnote-ref-3)