

OpenMRS 1.6 Usability Report

13 October 2010

Michael J. Downey
Patrick Proctor
Gretchen Tilley



SCHOOL OF INFORMATICS

INDIANA UNIVERSITY

IUPUI

TABLE OF CONTENTS

Table of Contents	2
Index of Figures	3
Executive Overview	4
Application Overview	5
Methods	7
Scenario-based Inspection	7
Heuristic inspection.....	8
Results	10
User Type: Registration Clerk	11
Issue A-1: Patient ID Generation	11
Issue A-2: Java Error Recovery.....	12
Issue A-3: In-Application Help	13
Issue A-4: Form Field Contextual Help	13
Issue A-5: Advanced Name-Based Searching.....	14
Issue A-6: Edit Patient Form Choices	14
Issue A-7: Verbose Patient Search	15
Issue A-8: Header Section Identification.....	17
Issue A-9: Link Colors	17
Issue A-10: Patient vs. Person	18
User Type: Data Entry Clerk	19
Issue B-11: Entering Data from Dashboard.....	19
Issue B-12: Technical Jargon	19
Issue B-13: Concept Searching	20
Issue B-14: Form Validation Error Language.....	20
Issue B-15: Patient Scheduling	21
Issue B-16: Moving Between Multiple Patients.....	22
User Type: Researcher	22
Issue C-17: Cohort Creation and Reports	22
Issue C-18: Multivariate Cohorts	23
Issue C-19: Base Concept Dictionary Content.....	24
Issue C-20: Cohort UI Inconsistency.....	25
User Type: Clinician	25
Issue D-21: Patient ID Generation.....	25
Issue D-22: Relationship Between Patient, Encounter, & Observation	25
Issue D-23: Non-Drug Order Entry	26
Issue D-24: Recent Data on Dashboard	28
Issue D-25: Demographic Data Display	28
Issue D-26: Default Longitudinal Reports.....	29
Issue D-27: System Name Consistency	29
Recommendations	31
Key issues.....	31
Next steps	31
Appendices	33
Scenarios and Task Lists	33

INDEX OF FIGURES

Figure 1. Patient ID Generation (Issue A-1).....	11
Figure 2. Java Error Recovery (Issue A-2)	12
Figure 3. In-Application Help (Issue A-3).....	13
Figure 4. Form Field Contextual Help (Issue A-4).....	14
Figure 5. Edit Patient Form Choices (Issue A-6).....	15
Figure 6. Verbose Patient Search (Issue A-7).....	16
Figure 7. Header Section Identification (Issue A-8)	17
Figure 8. Link Colors (Issue A-9)	18
Figure 9. Patient vs. Person (Issue A-10)	18
Figure 10. Technical Jargon (Issue B-12)	19
Figure 11. Concept Searching (Issue B-13)	20
Figure 12. Form Validation Error Language (Issue B-14).....	21
Figure 13. Patient Scheduling (Issue B-15)	21
Figure 14. Moving Between Patients (Issue B-16)	22
Figure 15. Cohort Creation and Reports (Issue C-17)	23
Figure 16. Multivariate Cohorts (Issue C-18)	23
Figure 17. Base Concept Dictionary Content (Issue C-19).....	24
Figure 18. Relationship Between Patient, Encounter, & Observation (Issue D-22)	26
Figure 19. Non-Drug Order Entry (Issue D-23)	27
Figure 20. Recent Data on Dashboard (Issue D-24)	28
Figure 21. Demographic Data Display (Issue D-25)	28
Figure 22. Default Longitudinal Reports (Issue D-26).....	29
Figure 23. System Name Consistency (Issue D-27)	30

EXECUTIVE OVERVIEW

We conducted a usability evaluation of OpenMRS, an open source electronic medical record system used primarily in the developing world. This system is an extensible platform, so we restricted our usability test to the basic “out of the box” configuration of the application.

Scenarios were developed based on existing documentation and user interviews. The system was evaluated both against these scenarios and resulting goals and tasks. We also conducted an expert heuristic review against standardized design criteria.

The results are presented here ranked by severity and sorted by user type developed during the course of the research. Justification and recommendations are provided for each issue discussed.

Finally, we offer some guidance in prioritization of the issues discovered in this phase of our work. Our next steps are to conduct user testing to validate these findings in situ, as well as potentially develop further areas for improvement of OpenMRS in future releases.

APPLICATION OVERVIEW

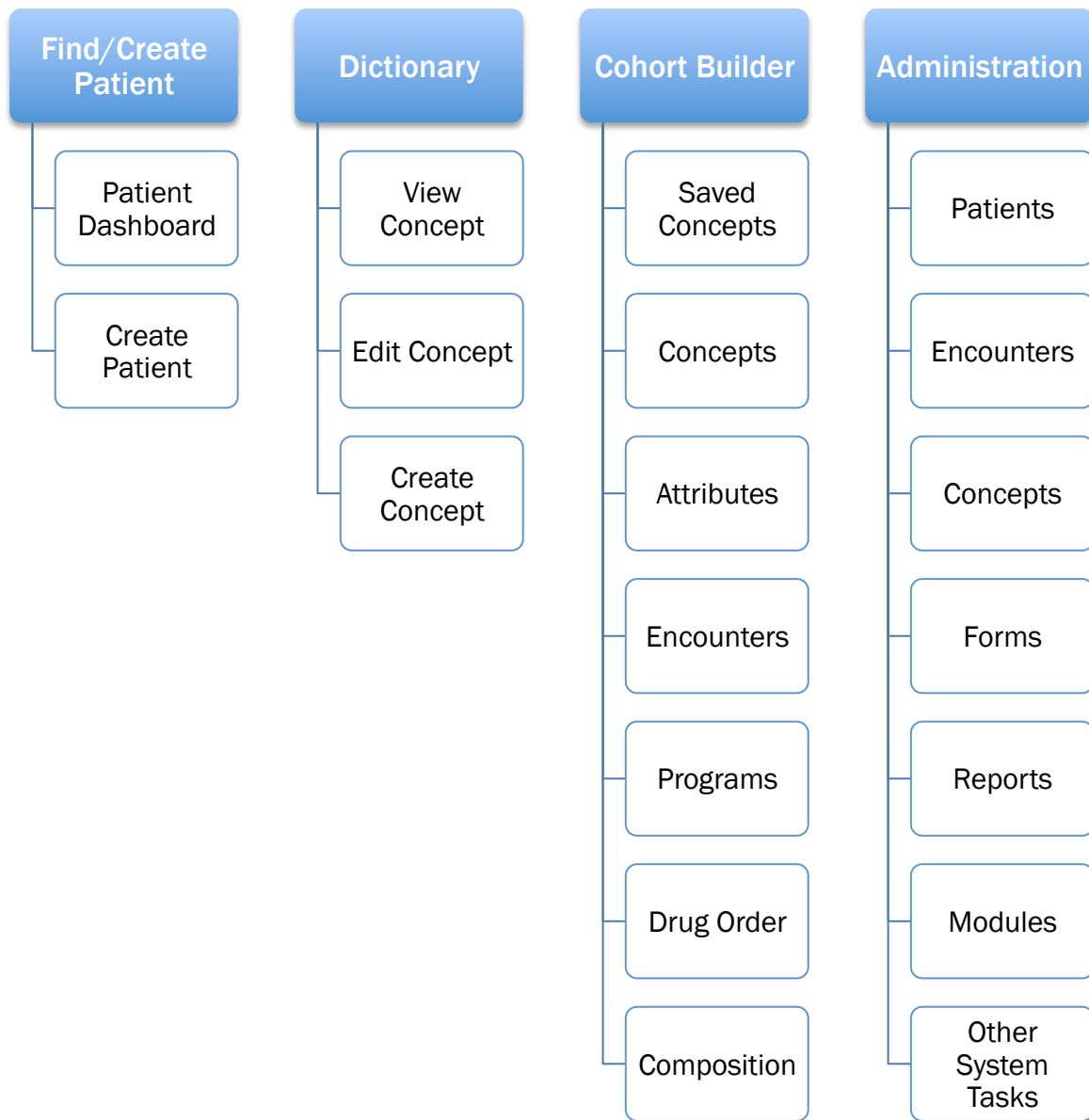
OpenMRS (Open Medical Records System) is an open source project that was started at Regenstrief Institute, an affiliated organization with the Indiana University School of Medicine. Its genesis came from the need to better manage medical records to combat the spread of HIV in Africa. Today, it is a general-purpose medical record management system, able to run on Mac OS X, Linux, and Windows.

Because of OpenMRS' wide variety of operating environments, it must be extremely flexible and open. It can be run on a laptop or server, on top of several operating systems. It must be multilingual and have UI design that takes into account the varied background of its users.

At its core, the application allows management of patient observations of various concepts (tests, demographics, vital signs, etc.) that are curated in a user-managed concept dictionary.

However, along with this flexibility comes a somewhat basic “out of the box” application. OpenMRS hosts an extensive collection of community-developed add-on modules that further customize the system for use in specific contexts, environments, and for different types of health care. Many of these modules have been developed in the field by users of the system to fit their own needs, and then shared with others around the world.

The following figure represents a simplified structural map of OpenMRS 1.6.1. After logging in, the system is divided into four main sections as shown. Both the “Find/Create Patient” section and “Dictionary” section are fairly straightforward and allow the user to work with patient data or manage the concept dictionary. The “Cohort Builder” section allows the user to define cohort groups of patients, which can then be used in creation of basic reports. The “Administration” section contains not only links to system administration tasks, but also includes data and information administration tasks as well.



METHODS

Key stakeholders of the OpenMRS leadership team were consulted at the beginning of the project between 29 and 30 August 2011, to ascertain major types of users of the systems and tasks those users might accomplish. The results of that survey were merged with existing OpenMRS design documentation on their web site and past user interviews to create five types of users: registration clerk, data entry clerk, clinician, researcher, and system administrator.

Note: Due to various factors including time and available resources, this study has de-scoped system administrator from the scenarios studied. Further research into the tasks and system design for this role will be valuable to the overall design of OpenMRS.

After fully developing the four testing user types into personas, we conducted a full scenario-based inspection of OpenMRS 1.6.1 for each user type. The public demo of this version of the application (the latest stable release) at demo.openmrs.org was used for all evaluation work. Following the scenario-based inspection, we conducted a heuristic inspection of the application against generally accepted criteria.

Scenario-based Inspection

Based on existing data and stakeholder feedback, we developed the usage scenarios and goals for the following four prototypical OpenMRS users:

1. **Scenario:** Register a new patient
 - a. **User:** Gilbert Leonis, registration clerk, 24 years old
 - b. **Biography:** Born just outside Eldoret, Kenya and lived with his family on their farm through secondary school. He got a job with Moi University in Eldoret doing data entry, and was recently seconded to work in the AMPATH building next door, although the University still pays him. His primary responsibility is to register new patients to the program and re-issue identification cards to current patients. He is still single and has no children, and likes to go out to Signature night club most evenings until the early morning hours.
 - c. **Goal:** A new patient's demographic data is collected, entered in the system, and an identification card is issued.
2. **Scenario:** Enter data from a completed patient form
 - a. **User:** Damarice Langat, data entry clerk, 28 years old
 - b. **Biography:** Born in the village of Burnt Forest, Kenya. She attended secondary school near Burnt Forest and now works for the USAID-AMPATH Partnership in nearby Eldoret as a data entry clerk. Her primary role is to handle data entry for clinical forms for the HIV/AIDS prevention programme. She became interested in computers in high school but had no formal IT training other than what has been provided

to her at AMPATH. She is married with three children, aged 8, 10, and 11. She does not have a computer at home, but occasionally uses one in town to access Facebook and e-mail.

- c. **Goal:** Patient data provided on a paper form filled by a clinician is entered and stored in the system.
3. **Scenario:** Review specific patient history
 - a. **User:** Dr. Wendell Brown, Clinician-doctor specializing in pediatrics; 34 years old
 - b. **Biography:** Dr. Brown is a volunteer with Doctors Without Borders and works three months each year in Kenya caring for children and adults with HIV/AIDS. He immigrated from Nicaragua as a child and is now a naturalized citizen of the U.S.A. with an American name. He received his medical training from John Hopkins University. He is comfortable using computers to input information, review files, and order tests, treatments, x-rays and prescriptions. OpenMRS is used in the clinics where he sees patients. He uses the software to access patient records and review past treatments, test and x-ray results, existing conditions, and past diagnosis. He often reviews the patient file again after the information has been input by the data entry clerk to add any additional notes and to review what was entered.
 - c. **Goal:** Retrieve patient information to review past examinations and diagnoses as well as the results of tests, x-rays and treatments to provide the best care possible.
4. **Scenario:** Generate reports on a group of patients
 - a. **User:** Robert Mullin, researcher, 43
 - b. **Biography:** Robert lives in Chicago, Illinois. He uses the software to track occurrences of various illnesses in discrete geographic areas where OpenMRS is used. He generally views computers as business machines and does not have much interest in using them outside of the workplace.
 - c. **Goal:** Patient demographic data is analyzed and used to track the spread of different diseases in eastern Africa or other broad geographical areas.

Heuristic inspection

After evaluating the above scenarios and documenting problems encountered while executing them, we then analyzed OpenMRS against well-known heuristics of design, based upon well-known standards created Jakob Nielsen (CITE) and extensively studied in academic literature and research. The following 10 heuristics were evaluated:

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom

4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Help users recognize, diagnose, and recover from errors
8. Help and documentation
9. Visual clarity
10. Important information belongs above the fold

Problems identified in the scenario-based inspection and heuristic inspection were then synthesized into a unified list, and sorted into one of the heuristic categories above.

RESULTS

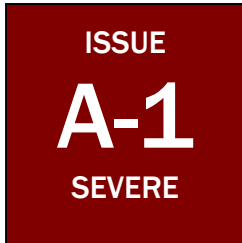
As a result of the scenario-based and heuristic inspections described above, we identified 27 usability problems with the base OpenMRS web application. They are presented here grouped by the four user types developed in the analysis process.

Severity terms used in this document, and their meaning, include:

- **Severe:** An urgent issue that may cause loss of data or prevents continued use of the system. (4 issues)
- **High:** A serious condition that make the system very difficult to use for many users or requires “workarounds” to accomplish tasks. (8 issues)
- **Medium:** Data is not lost, but this problem causes users to become confused or annoyed, and may cause them to not use the system as designed. (11 issues)
- **Low:** Cosmetic, minor design, or trivial issues that some users may find annoying. (4 issues)

It is important to remember that severity represents only the potential impact to users, not the importance to OpenMRS design strategy. Many “Low” severity issues are easy to fix, and many “Severe” issues would be very effort-intensive. Priority should be determined based on existing obligations and design plans, as well as the information presented in this report.

User Type: Registration Clerk



Patient ID Generation

- The system requires a correctly-generated patient ID number, but has no option to generate one. Users will not be able to register a new patient if either a list of ID numbers is not available, or they do not know how to generate one.
- Problem Type: Match between system and the real world
- Recommendation: Allow an option for the instant generation of valid, random, unique patient ID numbers when a new patient is created, but allow the possibility for such numbers to be overridden if necessary.

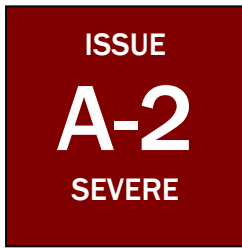
❌ Patient identifier check digit is invalid

Create a New Patient

Name	Given	Middle	Far
	<input type="text" value="Test"/>	<input type="text"/>	<input type="text" value="Use"/>

ID Number(s)	Identifier	Identifier Type
	<input type="text" value="424242"/>	<input type="text" value="OpenMRS Identification Number"/>
	<input type="text"/>	<input type="text" value="Old Identification Number"/>

Figure 1. Patient ID Generation (Issue A-1)

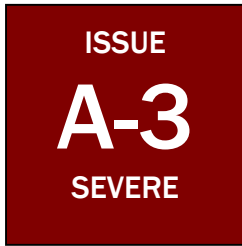


Java Error Recovery

- Some Java errors show no context of use. When users see error messages with Java stack traces, they will not necessarily know what they were doing when the error occurred, or how to avoid having it happen again.
- Problem Type: Help users recognize, diagnose, and recover from errors
- Recommendation: Evaluate options to handle Java exceptions with a short description of what the system was doing when the error occurred. Continue to minimize stack trace errors to avoid overwhelming users. Provide an opportunity for users to submit a "crash report" about the error, and basic instructions (perhaps with a link) about how to continue working.



Figure 2. Java Error Recovery (Issue A-2)



In-Application Help

- No in-application help documentation whatsoever. Without basic help, users will likely become confused about how to perform basic tasks in the system and may not be able to use the system at all.
- Category: Help and documentation
- Recommendation: Create basic in-application help documentation for common tasks, perhaps starting with the scenarios listed in this work.

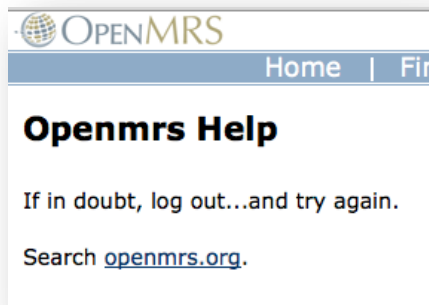
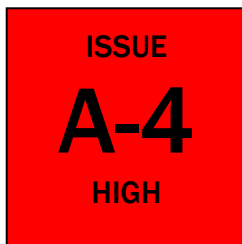


Figure 3. In-Application Help (Issue A-3)



Form Field Contextual Help

- Hover "tool tip" pop-ups to explain data entry fields are implemented inconsistently across forms and sometimes not present, and no icon is present to inform the users more information is available. Users may be uncertain about the meaning of form field titles and enter incorrect data.
- Category: Help and documentation
- Recommendation: Review all data entry forms in the application, and add both hover-based tool tips that describe each field, and add a small icon for the users to point to for the meaning.

Editing Concept WEIGHT (KG)

[Previous](#) | [View](#) | [Stats](#) | [Next](#) | [New](#)

Id 5089

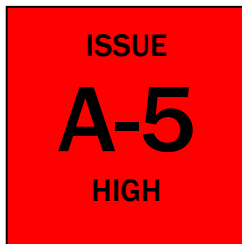
Locale [English](#) | [Spanish](#) | [French](#) | [Italian](#) | [Portugu](#)

Name

Short Name

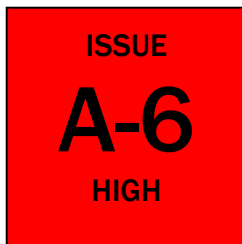
Description The text shown in report headers and other places where space is limited

Figure 4. Form Field Contextual Help (Issue A-4)



Advanced Name-Based Searching

- No soundex-based searching for names. Names must be spelled exactly correct for a match. There is no "advanced search" that looks for other identifiers other than name or ID number. As a result, some users may not find the correct patient in the system and create a duplicate record.
- Category: Error prevention
- Recommendation: Investigate search methods for foreign names and alternative search algorithms that will return similar sounding names, omitted double letters, missing vowels, etc.



Edit Patient Form Choices

- Two options for editing patient demographics, "Edit this Patient" and "Edit this Patient (Short Form)" exist. The longer form seems to duplicate all the information of the short form, making the short form redundant. Users may choose the short form and not record all information.
- Category: Match between system and the real world
- Recommendation: Unify the "short form" and regular edit patient form so only one option exists. If advanced options are desired but not required, indicate which fields are required, or "minimize" the extended attributes allowing the user to expand that section if desired before submitting the form.

John Mamwa Shavola
 53 yrs (~Dec 11, 1956)

BMI: ? (Weight: 52.5 kg , Height:) CD4: 296.0 | Regimen:
 Last encounter: **ADULTRETURN @ Unknown Location, Jun 6, 2006**

[Overview](#) | [Regimens](#) | [Encounters](#) | [Demographics](#) | [Graphs](#) | [Form Entry](#)

Patient

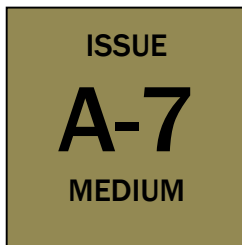
Names
 John Mamwa Shavola

Addresses

Address	Section/Homestead	Estate/Nearest Centre	Town/Village	Sublocation	Location	Division
			Lwandeti			

[Edit this Patient](#) | [Edit this Patient \(Short Form\)](#)

Figure 5. Edit Patient Form Choices (Issue A-6)



Verbose Patient Search

- The purpose of the checkbox labeled "Include Verbose" when searching (e.g. patient search, concept search) is unclear. When turning it on and off after a set of results appears, nothing seems to happen. A user may become confused about the box.
- Category: Match between system and the real world
- Recommendation: Change the description for this checkbox to be more obvious, or remove the checkbox entirely if it is not necessary.





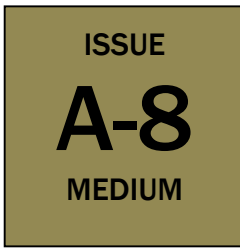
Find Patient(s)						
Patient Identifier or Patient Name: <input type="text" value="mamwa"/>						Include Verbose <input checked="" type="checkbox"/>
<i>Results for "mamwa". Viewing 1-4 of 4</i>						
Identifier	Given	Middle	Family Name	Age	Gender	Birthdate
1. 4450MP-4	Dona	Mamwa	Chepkorir	43		09/26/1967
2. 7164AM-7	Gedeon	Mamwa	Omboya	31		09/02/1979
3. 7279MP-4	John	Mamwa	Shavola	53		≈ 12/11/1956
4. 8751MO-3	Venny	Mamwa	Berich	33		12/21/1976
Add New Patient						

Figure 6. Verbose Patient Search (Issue A-7)



Header Section Identification

- Items in header menu don't respond to current section, making it difficult to know which section user is in. As a result, the user can not tell which section (patient, administration, dictionary) they are working within.
- Category: Visibility of system status
- Recommendation: Consider changing the visual status of the header menu to represent what logical section each page belongs to, allowing the user to easily correlate a form or other system page with that section.

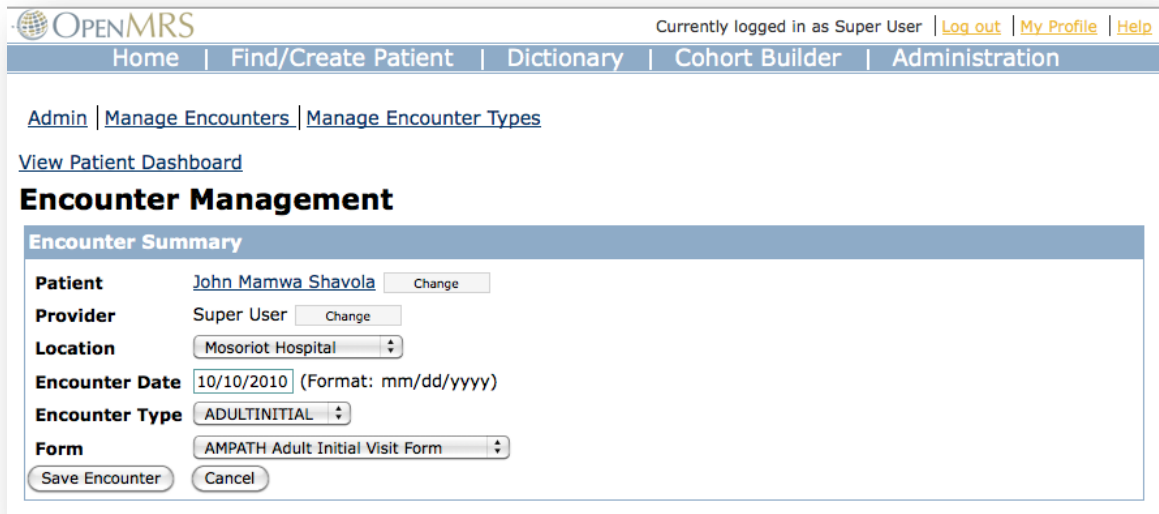
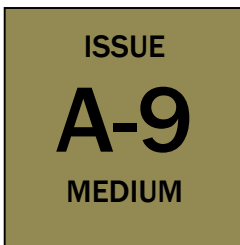


Figure 7. Header Section Identification (Issue A-8)



Link Colors

- Inconsistent link coloring. Some items appear to be links but are not. Users may try to click something that is not a link, or avoid clicking something that is in fact a link, causing them to become confused how to edit a value or make changes to the system. *Example:* Blue text is used both as link color and highlighted text color. User-action links (log out, profile, help) are yellow.
- Category: Visibility of system status
- Recommendation: Review stylesheets and other design elements to standardize link color and presentation. Do not use those colors for other design elements to allow links to "stand out" on pages.

John Mamwa Shavola
 53 yrs (~Dec 11, 1956)
BMI: ? (Weight: 52.5 kg , Height:) CD4: 296.0 | Regimen:
 Last encounter: **ADULTRETURN @ Unknown Location, Jun 6, 2006**

Overview | Regimens | Encounters | Demographics | Gr

Patient

Figure 8. Link Colors (Issue A-9)

ISSUE
A-10
 LOW

Patient vs. Person

- Inconsistent use of the word "patient" and "person" within the application. Some users may be confused if they are dealing with a patient or a system user.
- Category: Consistency and standards
- Recommendation: Review instances of such language and when it refers to a patient, change the language.

Create Patient

To create a new person, enter the person's name and other information below first to double-check that they don't already have a record in the system.

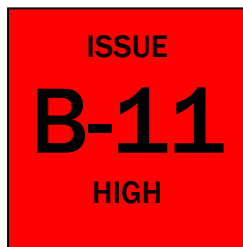
Name

Birthdate or Age
 (Format: mm/dd/yyyy)

Gender Male Female

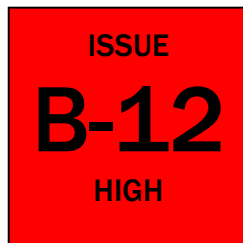
Figure 9. Patient vs. Person (Issue A-10)

User Type: Data Entry Clerk



Entering Data from Dashboard

- Cannot enter patient data while viewing patient information. The system works in only a data-entry or data-viewing mode. There is no way to compare past values while entering new observations, which would allow the user to ensure the entered value is likely accurate and not mistakenly written or abnormal for that particular patient.
- Category: Match between system and the real world
- Recommendation: Allow form entry within the web application, with the most recent form's value on one side of the screen, and the new form on the other side of the screen, allowing the user entering data to see the previous values of the same form, from the last encounter.



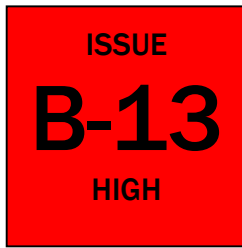
Technical Jargon

- There is widespread use of programming terminology throughout the web application. Examples: "type 4 (INVALID CHECK DIGIT) if hasCheckDigit is flagged". Users will not understand this language and may abandon what they are doing due to confusion.
- Category: Match between system and the real world
- Recommendation: Review error and warning messages for jargon and reword them using plain language.

Observation

Person	John Mamwa Shavola <input type="button" value="Change"/>
Encounter	Unknown Location - Jun 6, 2006 View/Edit
Order	<input type="text"/>
Location	<input type="text" value="Unknown Location"/>
Observation Date	<input type="text" value="06/06/2006"/> (Format: mm/dd/yyyy)
Concept	WEIGHT (KG) <input type="button" value="Change"/> Patient's weight in kilograms.
Value	<input type="text" value="424242"/> kg Invalid value for numeric:
Comment	<input type="text"/>

Figure 10. Technical Jargon (Issue B-12)

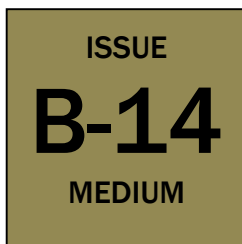


Concept Searching

- When searching for a concept to enter its value, the user must know the concept's name in the dictionary and type at least the first 3 letters correctly before it appears. Otherwise, the results are blank. The user must know the name of the concept they are looking for before entering it.
- Category: Recognition rather than recall
- Recommendation: Consider "auto-complete" style UI elements that allow the user to see available choices as he types. Start offering these choices with the first letter rather than after the 3rd letter.

A screenshot of a web interface titled "Concept Dictionary Maintenance". At the top, there is a link: "Download the concept dictionary in CSV format -- (dynamically creates a CSV file containing)". Below this is a search section with a blue header "Find Concept(s)". It contains a search input field with the text "TB", followed by two checkboxes: "Include Retired" (checked) and "Include Verbose" (checked). Below the input field, a message reads: "You must have at least 3 search characters".

Figure 11. Concept Searching (Issue B-13)



Form Validation Error Language

- Validation errors/warnings are often not written in plain language but rather application code. Users may not understand the problem with their form entry, and may not be able to recover from the error by correcting their data entry.
- Category: Error prevention
- Recommendation: Review all form validation messages and rewrite the language as necessary to explain the problem and how to avoid it.

Encounter Summary

Patient [John Mamwa Shavola](#)

Provider Super User

Location Chulaimbo

Encounter Date 06/dec/10 (Format: mm/dd/yyyy) encounterDatetime is an invalid Date

Encounter Type ADULTRETURN

Form v

Created By [Super User](#) - Jun 7, 2006

Voided

Figure 12. Form Validation Error Language (Issue B-14)

ISSUE
B-15
MEDIUM

Patient Scheduling

- The system provides no scheduling system and no obvious way to record next appointment other than as an encounter's observation, which is not easily reportable and not shown on the patient dashboard. Users will have to maintain an external patient scheduling system.
- Category: Match between system and the real world
- Recommendation: Allow the system to show the value of a most recent observation for a “next appointment” observation or create a basic scheduling system integrated into the core application.

John Mamwa Shavola
53 yrs (b: Dec 11, 1956)

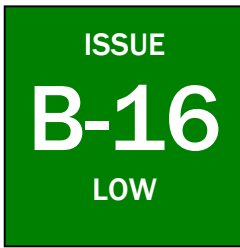
ADULTRETURN @Unknown Location | 06/06/2006 | Super User

-- RETURN VISIT DATE: **Jun 20, 2006**

-- PRIMARY REGIMEN: **true**

-- PAY CATEGORY: **PILOT**

Figure 13. Patient Scheduling (Issue B-15)



Moving Between Multiple Patients

- No easy way to move between patients without going back to the search page. As a result, users have to perform additional tasks to search for users.
- Category: User control and freedom
- Recommendation: Create a “smart search box” similar to the OpenMRS.org wiki search, shown in Figure 14, and place it at the top of every screen. This search should include both patients and the concept dictionary.

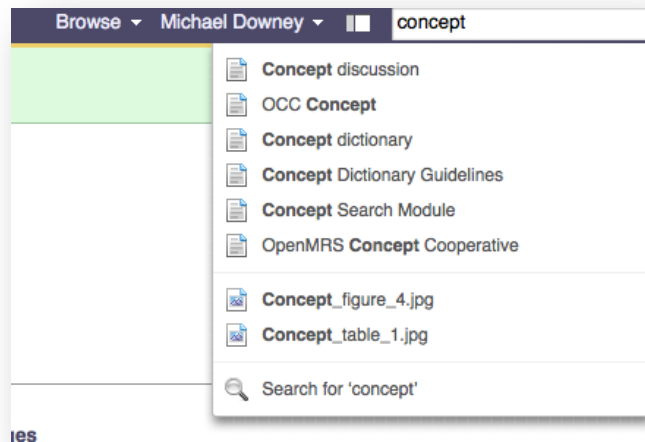
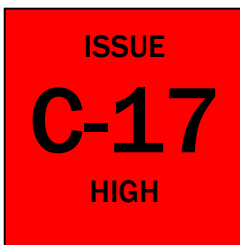


Figure 14. Moving Between Patients (Issue B-16)

User Type: Researcher



Cohort Creation and Reports

- Sequence of cohort creation and report creation is not clear. A new user will become confused and will likely not understand how to identify cohorts and create reports from them.
- Category: User control and freedom
- Recommendation: Consider creating a "report creation wizard" that guides the user through the process of selecting and saving a cohort, as well as a similar wizard to create basic reports.

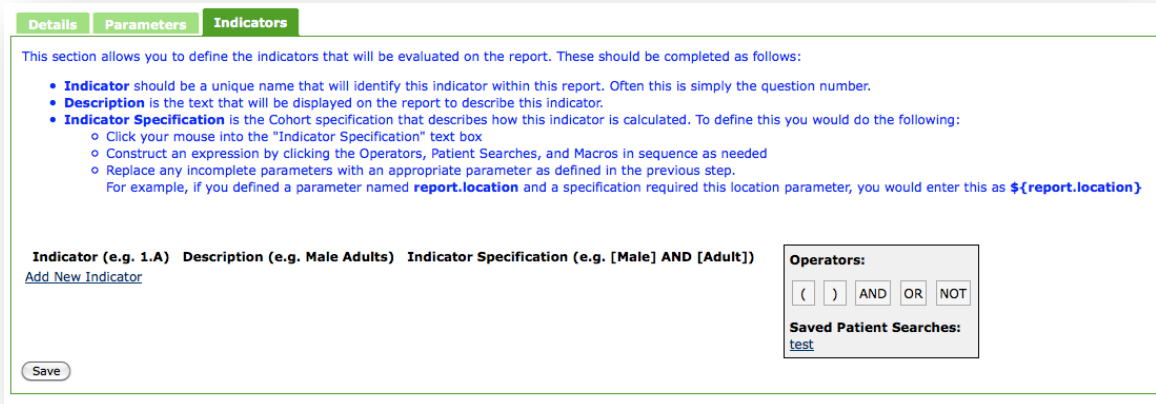
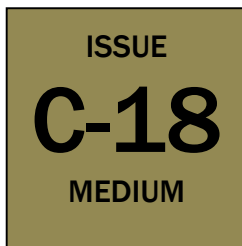


Figure 15. Cohort Creation and Reports (Issue C-17)



Multivariate Cohorts

- In Cohort Builder, there is no way to create a multivariate cohort. (e.g., Combining demographics and person attributes, or program enrollment with drug orders.) This may cause a user to create multiple simple cohorts instead of one complex group.
- Category: Match between system and the real world
- Recommendation: Consider a single form rather than a tabbed interface that allows the selection of multiple criteria in different “categories”.

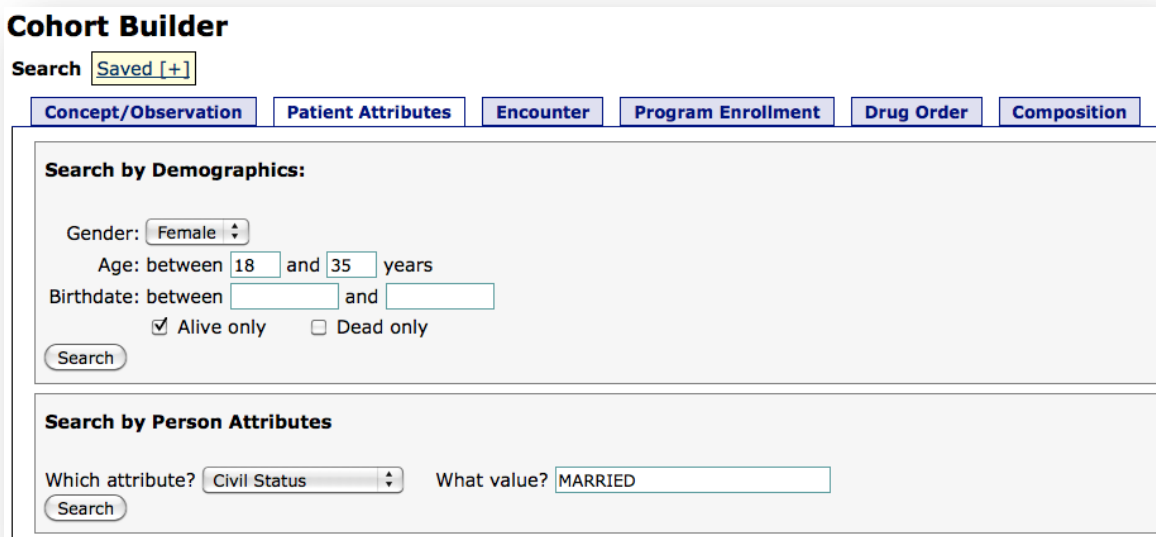
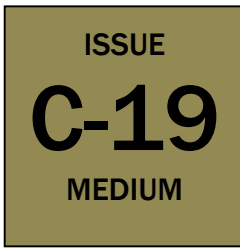


Figure 16. Multivariate Cohorts (Issue C-18)



Base Concept Dictionary Content

- Concept names are entered in all capital letters, which is more difficult to scan and read. This may be annoying for users that read many lists of concepts during a day.
- Category: Visual Clarity
- Recommendation: Revise the default concept dictionary to use mixed case.

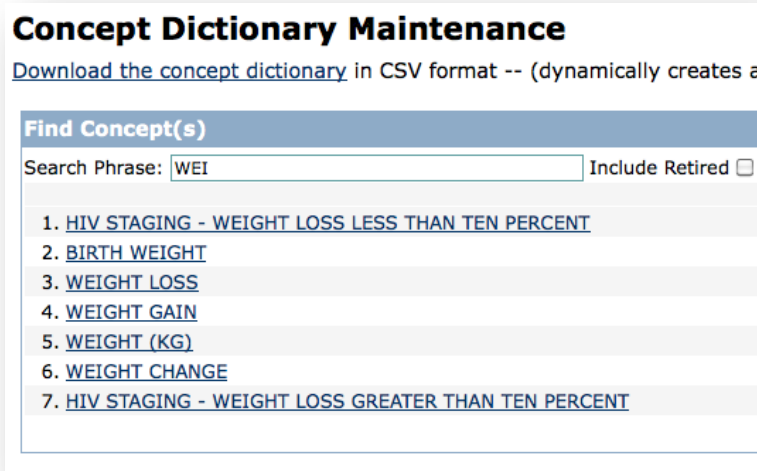
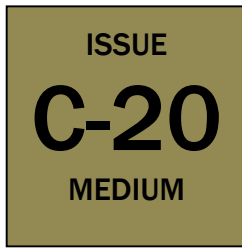


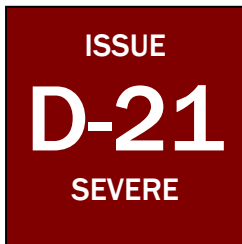
Figure 17. Base Concept Dictionary Content (Issue C-19)



Cohort UI Inconsistency

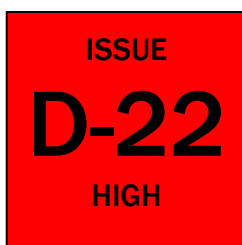
- UI design for cohorts and reporting, patient dashboard, and data entry are not consistent. Users will have to "learn" the layout of each tool with extra effort because the flow through these tools is not the same.
- Category: Consistency and standards
- Recommendation: Redesign the cohort, reporting, and encounter management tools to use a similar layout and design as the patient dashboard, or otherwise unify the design of these tools to use same-looking widgets, forms, fields, etc.

User Type: Clinician



Patient ID Generation

- No intuitive way for a clinician to enter an encounter form when viewing a patient. No error message when searching for a form from the patient dashboard. This can only be accomplished via the system administration section.
- Category: Match between system and the real world
- Recommendation: Create a link from the patient dashboard to allow creation of a new encounter. The same forms and pages should be used as those that are already existing from the administration section, to avoid additional confusion of multiple methods to enter forms.



Relationship Between Patient, Encounter, & Observation

- No visual relationship between patient, encounter, and observations. From the patient dashboard, a user can only see encounter forms and some selected recent observations.
- Category: Visibility of system status
- Recommendation: Alternate views, such as a column based display could show the relationship between encounters and observations, and the values of those observations. Consider alternative layouts.

John Mamwa Shavola
 53 yrs (~Dec 11, 1956)

BMI: ? (Weight: 52.5 kg , Height:) CD4: 296.0 | Regimen:
 Last encounter: **ADULTRETURN @ Unknown Location, Jun 6, 2006**

Overview | Regimens | Encounters | Demographics | Graphs | Form Entry

Encounters

Show 20 entries

Edit	View	Encounter Date	Encounter Type	Provider	Form	Location	Enterer
		06/06/2006	ADULTRETURN	Super User		Unknown Location	Super User
		05/09/2006	ADULTRETURN	Super User		Unknown Location	Super User
		04/11/2006	ADULTRETURN	Super User		Unknown Location	Super User

Showing 1 to 3 of 3 entries

Figure 18. Relationship Between Patient, Encounter, & Observation (Issue D-22)

ISSUE
D-23
 MEDIUM

Non-Drug Order Entry

- There is no way to add orders or regimens that are not drug-based. Users will have to create observations within encounters for such orders that are not able to be seen without viewing individual encounter forms.
- Category: Match between system and the real world
- Recommendation: Bring consistency to drug vs. non-drug orders by tying drug regimens to an encounter's observation, or allowing other non-drug concepts, e.g., X-RAY as regimens (perhaps reworded to orders) visible from the patient dashboard.

John Mamwa Shavola
OpenMRS Identification Number: **7279MP-4**

53 yrs (~Dec 11, 1956)

BMI: ? (Weight: 52.5 kg , Height:) CD4: 296.0 | Regimen:

Last encounter: **ADULTRETURN @ Unknown Location, Jun 6, 2006**

Overview
Regimens
Encounters
Demographics
Graphs
Form Entry

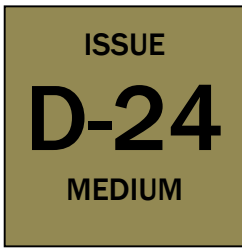
Current and Future Regimens

Item ordered	Dose/Units	Frequency	Start date	Scheduled Stop Date	Instructions
<i>(No orders)</i>					
<i>(No orders)</i>					
OTHER REGIMENS					
<i>(No orders)</i>					
(+ Add/Change Regimen)					
Add a standard drug regimen 3TC + d4T(30) + NVP (Triomune-30) 3TC + d4T(40) + NVP (Triomune-40) AZT + 3TC + NVP AZT + 3TC + EFV(600) d4T(30) + 3TC + EFV(600) d4T(40) + 3TC + EFV(600)			Add your own drug regimen Drug <input style="width: 80px;" type="text"/> Dose <input style="width: 60px;" type="text"/> Frequency <input style="width: 60px;" type="text"/> 1/day x <input style="width: 60px;" type="text"/> 7 days/week Start date <input style="width: 80px;" type="text"/> (mm/dd/yyyy)		

Completed Regimens

Item ordered	Dose/Units	Frequency	Start date	Scheduled Stop Date	Actual Stop Date	Instructions	Reason discontinued
<i>(No orders)</i>							
<i>(No orders)</i>							
OTHER REGIMENS							
<i>(No orders)</i>							

Figure 19. Non-Drug Order Entry (Issue D-23)



Recent Data on Dashboard

- Last encounter and most recent observations (CD4, BMI, Weight, Height, Regimen) are shown on patient dashboard but not linked to that encounter. Users will have to browse through individual encounters to find the most recent encounters with this information.
- Category: User control and freedom
- Recommendation: Create web links from recent encounter values to that encounter, e.g., CD4, Weight, Height, Regimen, Last Encounter.

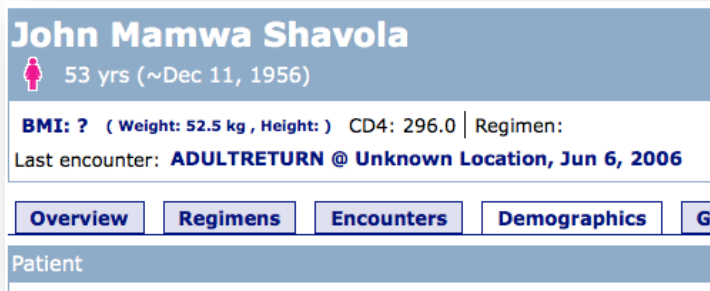


Figure 20. Recent Data on Dashboard (Issue D-24)



Demographic Data Display

- Some key patient demographic data is hidden behind a tab, causing users to search for such information through tabs and individual encounter data.
- Category: Important Information Belongs Above the Fold
- Recommendation: Consider moving demographic data to the Overview tab.

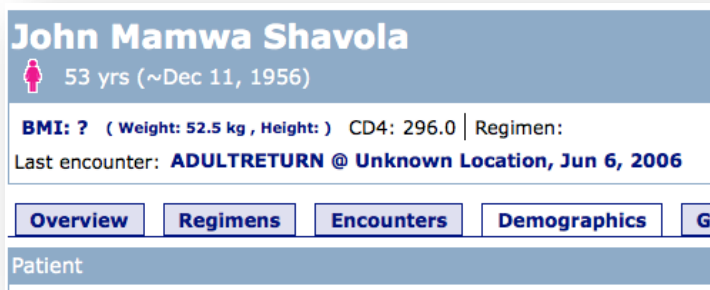
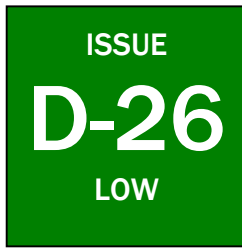


Figure 21. Demographic Data Display (Issue D-25)



Default Longitudinal Reports

- Longitudinal reports must be created per patient, no standardized chart/graph is provided for summary data. Users must create charts for each patient when viewing their patient dashboards.
- Category: User control and freedom
- Recommendation: Design the default patient dashboard to include longitudinal data graphs for basic information shown on the patient dashboard, e.g., BMI, Weight, Height, CD4.

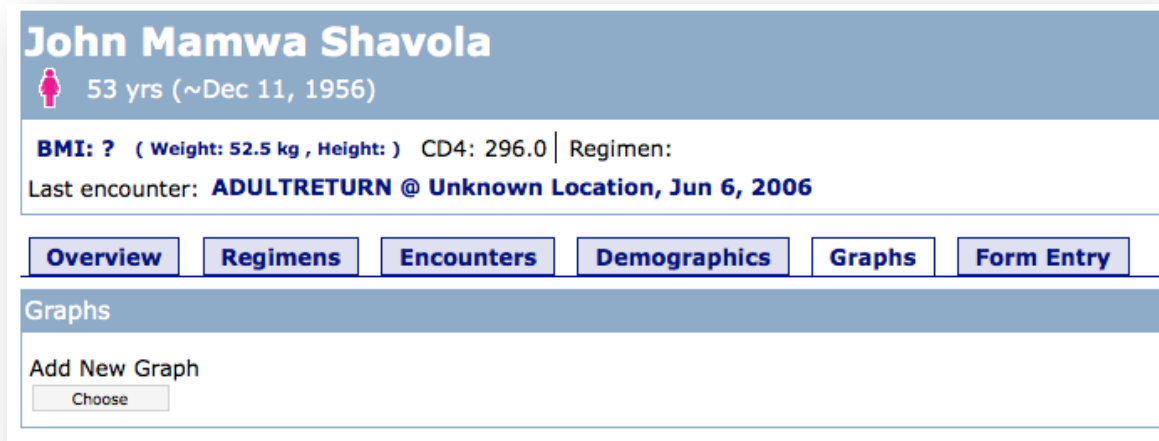
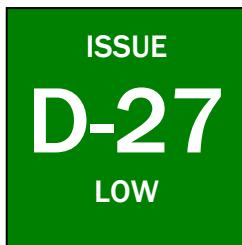


Figure 22. Default Longitudinal Reports (Issue D-26)



System Name Consistency

- System name (OpenMRS) is not presented consistently (e.g. Openmrs). Some users may be distracted by this inconsistency.
- Category: Consistency and standards
- Recommendation: Review all instance of application name throughout the application and modify for consistency.



Figure 23. System Name Consistency (Issue D-27)

RECOMMENDATIONS

Key issues

This study identifies 4 severe, urgent issues which should be evaluated as soon as possible to ascertain priority within the OpenMRS community:

- **Issue A-1:** User ID and check-digit generation problems
- **Issue A-2:** Unrecoverable Java stack traces/errors
- **Issue A-3:** No in-app help screens
- **Issue D-21:** No encounter entry from patient dashboard

Based on the validity of the scenarios that we evaluated, these issues have the potential to affect data loss by users' misunderstanding of how to use OpenMRS, or inability to find certain functions.

Issues A-3 and D-21 seem particularly suited for “easy” implementation, as they likely require little application development and coding. Some OpenMRS help is available in the project's wiki, which could be included in the application help pages. Additionally, anything created for in-application help would be of value to people on the project wiki site as well, allowing newcomers to the project to understand more about the application's use.

Issues A-1 and A-2 will require additional development effort. Addressing Issue A-1 (ID numbers) may make the system easier to deploy in some contexts because it won't be necessary to have pre-generated ID numbers or an external number generator that would have to be used in parallel to OpenMRS. Issue A-2 will be helpful to all users of the system to recover from unexpected errors. Additional research on the OpenMRS project has indicated that in future versions of the software, these errors will produce a report that can be sent for analysis. This functionality should be increased to allow the user to easily “go back” to the previous state without losing any data entry that was happening.

All of the severity levels have some issues with trivial fixes, and some issues with substantial work required. Thus, the 8 “high” severity issues, as well as the medium and low severity issues should be reviewed for feasibility of implementation and scheduled for upcoming releases if appropriate.

Next steps

OpenMRS is an open source software project, and as such has been developed over several years by various individuals and organizations. This disparity is highly evident in its design, and many issues discovered in this evaluation point to that fact. Efforts to unify the design of various components of the core web application would make

the system feel much more “polished” and people will find the system easier to use due to common navigation elements, links, layout, etc.

Research into the OpenMRS project suggests that there are no central interface design guidelines in place, nor are there any user profiles that designers of the core application and add-on modules can use to ensure their applications conform to some basic user expectations and scenarios. We hope that the scenarios presented here can be a starting point for that work and user focus in the OpenMRS community as it grows.

OpenMRS is a complex application, designed to be extensively customized by nearly all users. This study set out to evaluate the “out of the box” OpenMRS system and ensure it is usable for very basic tasks that might be conducted in a clinic environment.

Additional studies are highly suggested to evaluate typical combinations of add-on modules used for OpenMRS and with additional scenarios.

APPENDICES

Scenarios and Task Lists

1. **Scenario 1:** Review specific patient history
 - a. **User:** Dr. Wendell Brown, Clinician-doctor specializing in pediatrics; 34 years old
 - b. **Biography:** Dr. Brown is a volunteer with Doctors Without Borders and works three months each year in Kenya caring for children and adults with HIV/AIDS. He emigrated from Nicaragua as a child and is now a naturalized citizen of the U.S.A. with an American name. He received his medical training from John Hopkins University. He is comfortable using computers to input information, review files, and order tests, treatments, x-rays and prescriptions. OpenMRS is used in the clinics where he sees patients. He uses the software to access patient records and review past treatments, test and x-ray results, existing conditions, and past diagnosis. He often reviews the patient file again after the information has been input by the data entry clerk to add any additional notes and to review what was entered.
 - c. **Goal:** Retrieve patient information to review past examinations and diagnoses as well as the results of tests, x-rays and treatments to provide the best care possible.
 - d. **Task 1:** In seeing a patient, Dr. Brown will access OpenMRS to review any previous visits by the patient. He wants to know past and present diagnosis, past and current medicines, patient medical history, and patient living situation (is clean water available, is electricity available, etc.)
 - e. **Task 2:** Record new patient information including new complaints and concerns, observations, and any diagnosis.
 - f. **Task 3:** Order any needed tests and x-rays through OpenMRS
 - g. **Task 4:** Review patient file via OpenMRS after data entry clerk inputs information.
2. **Scenario 2:** Generate reports on a group of patients
 - a. **User:** Robert Mullin, researcher, 43
 - b. **Biography:** Robert lives in Chicago, Illinois. He uses the software to track occurrences of various illnesses in discrete geographic areas where OpenMRS is used. He generally views computers as business machines and does not have much interest in using them outside of the workplace.
 - c. **Goal:** Patient demographic data is analyzed and used to track the spread of different diseases in eastern Africa or other broad geographical areas.
 - d. **Task 1:** Obtain the patient data *en masse* from OpenMRS' servers.
 - e. **Task 2:** Visualize the data by illness and specific geographical area.

- f. **Task 3:** Create a report that presents and interprets the OpenMRS data.
3. **Scenario 3:** Enter data from a completed patient form
 - a. **User:** Damarice Langat, data entry clerk, 28 years old
 - b. **Biography:** Born in the village of Burnt Forest, Kenya. She attended secondary school near Burnt Forest and now works for the USAID-AMPATH Partnership in nearby Eldoret as a data entry clerk. Her primary role is to handle data entry for clinical forms for the HIV/AIDS prevention programme. She became interested in computers in high school but had no formal IT training other than what has been provided to her at AMPATH. She is married with three children, aged 8, 10, and 11. She does not have a computer at home, but occasionally uses one in town to access Facebook and e-mail.
 - c. **Goal:** Patient data provided on a paper form filled by a clinician is entered and stored in the system.
 - d. **Task 1:** Review the paper form. Obtain patient's name and ID number from the paper form. Search for the patient in the system. (If they do not exist, a different scenario for creating that patient will be done.
 - e. **Task 2:** Create a new encounter in the system associated with the patient and provider listed on the paper form. Ensure the location where the encounter happened and the date are recorded in the system.
 - f. **Task 3:** For each item listed on the form, record the concept and its associated value (e.g., weight and 100 kg). Repeat this process for each observation on the form.
Note: These three tasks are repeated for each form to be entered.
 4. **Scenario 4:** Register a new patient
 - a. **User:** Gilbert Leonis, registration clerk, 24 years old
 - b. **Biography:** Born just outside Eldoret, Kenya and lived with his family on their farm through secondary school. He got a job with Moi University in Eldoret doing data entry, and was recently seconded to work in the AMPATH building next door, although the University still pays him. His primary responsibility is to register new patients to the program and re-issue identification cards to current patients. He is still single and has no children, and likes to go out to Signature night club most evenings until the early morning hours.
 - c. **Goal:** A new patient's demographic data is collected, entered in the system, and an identification card issued
 - d. **Task 1:** Meet with the patient and determine their identity.
 - e. **Task 2:** Review current patient records to see if the patient is already registered.
 - f. **Task 3:** If not registered, create a new patient record.
 - g. **Task 4:** Verbally collect demographic information from the patient and enter it into the new patient record.

- h. **Task 5:** Fill out an ID card with the patient's ID number and name, laminate it and provide to the patient.
Note: These five tasks are repeated for new patient.