COMP 1650

User Interface Design

Coursework Title

User Interface Design for web-based crowdsourcing application

Submission Date

12 November, 2014

Banner ID- *000860937*

TABLE OF CONTENT

1. Introduction

Crowdsourcing means the process of getting required work, services or idea from a huge amount of people rather than traditional ways and especially from online society. The word of crowdsourcing comes from the combination of ‘crowd’ and ‘outsourcing’.
The case study is to investigate the user interface design for crowdsourcing desktop application that is concerned with facial skin diseases aimed to use in facial clinics.

2. Literature Review

2.1. Guideline for design principles

Nielsen’s Usability Heuristics

- Visibility of system status

The system should always inform users what to do or what is happening with clear messages in right timing.

- Match the system with real world

The system should use the easy language of words, phrases that the similar to the used of user in real world, rather than technical terms. Make information look like natural and logical order.

- User control and freedom

Users often make mistake by choosing wrong functions or processes, system should have ‘emergency exit’ to go back from unwanted situation without going to complex dialogues. Support to enable ‘undo’ and ‘redo’ functions.

- Consistency and standards

The system should always use same words, same actions in same situations and follow platform conventions.

- Error prevention

Prevent possible errors beforehand is better than informative, understandable error messages. Check and reject possible error conditions and take a confirmation option from user before they commit.

- Recognition rather than recall

Reduce the users' memory for making actions, options and tasks. User may not remember information from one to another. Instructions for the use of system should be obvious and easy to retrieve.

- Flexibility and efficiency of use

The system should be used both beginner and expert user. Allow users to modify frequent actions.

- Aesthetic and minimalist design

Dialogues should not include too many information that is not necessary or inappropriate. Every additional unit of information compares with the appropriate units of information.

- Help user recognize, and recover from errors

Error message should be displayed in simple language, exactly indicate what is the problem and suggest an effective solution.

- Help and documentation

System can use easily with any help is the best but the system may need to have help and documentation like user manual. Easy to search any information about system, the specific step of task that to be carried out and focused on user's task.

3. Analysis of issues

3.1. UI design issues for desktop application

The issues for user interface design generally occur when software is developed. The system cannot be successful if the UI has many problems even the system processes are usable. For the system can effectively use, UI design problems must be solved. There are various types of UI design problems. The following are common problems of UI design.

1) Message
2) System loading time
3) Icons and bitmaps
4) Text font & Color
5) No feedback
6) Scrolling

1) Message

While user makes mistakes or giving information to user, the message box is displayed. The message box problems is occurred if message box dialogue is displayed in place where cannot get user attention, (for example Message box is displayed at the back of application window) or dialogue box is not responded. Even more, text in message box can cause the problem. The message box must give clear and exact information. Users hate to read too many information that is giving many details to user or explaining a complex troubleshooting solution within message box.

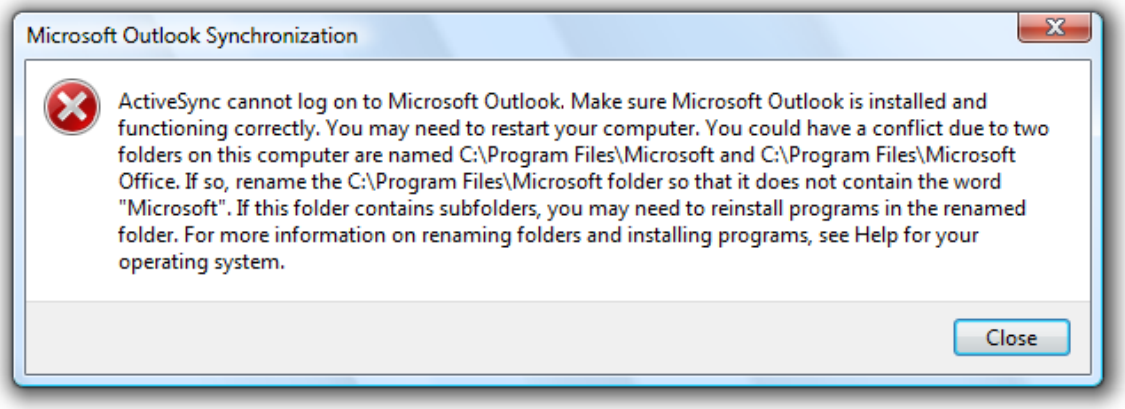


Figure 3.1.1

Another problem concerned with message box is that the error message is not given specific problem and how to solve. User cannot understand what is needed to do.

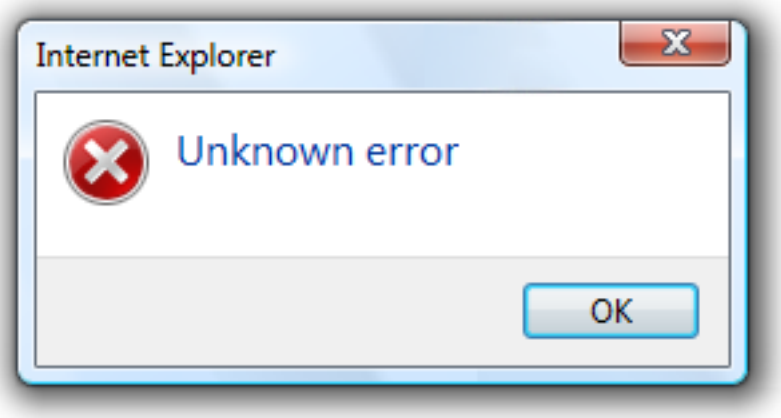


Figure 3.1.2

2) System loading is too long

This issue occurs loading time is too long when starting application or transaction one page to another. There are many reasons for this problem but concerned with UI design, using too many images or videos can cause to loading too long. So the application should not include images or videos rather than necessary.

3) Icons and bitmaps

User may misunderstand about the icons and bitmaps, which are wrong or not suitable with the real meaning. Icons and bitmaps usually depict certain functionality without using text. In this condition, users don't know how to use with unclear meaning of icon. So the application should label in some condition for user guideline. Example: to show text while mouse pointer is on icon.

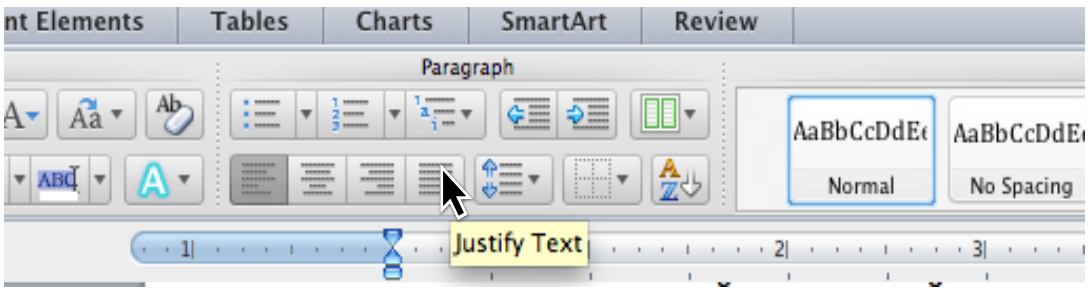


Figure 3.1.3

4) Text font and color

Using complex font style or handwriting font style in whole application tends to be inflexible design. Users don't want to use it if the text of the application is difficult to read. Next problem is about using color. The application uses light text color with light background or dark text color with dark background, a lack of contrast between text and its background will drive users insane. Their eyes don't know which color to focus on, which will almost instantly strain their eyes.

5) No Feedback

One of most basic issues is system is lack to give feedback. When system is taking to long to complete an action and it does not notify the user, the system is not a good one. Users think the action that they made is broken and try to start the new one. So the system should need to notify in some way, for example the process is taking more than a second, display the mouse cursor to be "busy".

6) Scrolling

User dislike to use horizontal scrolling and they always give feedback negatively when they meet it. On website, only vertical scrolling are expected by users. If the page has both horizontal and vertical scroll, users shall move their view into two dimensions. In contrast, only one scroll is a simple way to see content without any other planning.

3.2. Legal, Social or Ethical Issues of software development

Software development companies and the developers are often try to make their goal with illegal and unethical action because the problem of resources constraints that the developers face to make a quality product. There are many issues that faced by developers and software house. Some of these are as following.

- Using open source code in their system without crediting to original writer
- Using illegal software to achieve their tasks
- Making reverse process of find out how it works from engineering code
- Take altitude from the competition
- Do not addressing known bugs

Using open source code

Open source is source code available used to create the product and there are three types of open source, Licensed code, copyright or credit code and public domain. Among these three types, the first two types may cause the ethical issues to developer because they don't provide credit for using code. According to the Digital Media Copyright Act (DMCA), it can cause a major issue to developers who use the code without giving credit to original writer.

Using illegal software

The company is temping to use reproduced copy software or break the software license due to time and money problems. Some of the largest companies are illegally used in the past but some companies are continuing to use these.

Making reverse process

Reversing process is a confusing and complex process in the software development. Rather than any other issues, this issue frequently makes problems for developers and companies. Reversing is a process of decompiling an application to get original source code. However, many developers are used in practice of reversing process to find out how the code performed what action.

Do not addressing known bugs

Developers and companies are often skipping on quality assurance test because of deadlines is close. It becomes major errors are found in the software or the major errors are not been fixed because of not enough time to re-test. The problem of these errors makes largely losses for businesses and inconvenience for users.

3.3. Platform issues

The developer should think about compatibility before designing an interface. It is important to understand what are the capability and accessibility of platform that the developed software will be run. The developer should follow the accessibility guidelines of the development platform or operation system.

The system cannot successful if the design that the developer wants to make is not compactible with the platform run across that application. For example, generally the resolution of Apple platform (OS X, etc) and Microsoft Platform is not the same. And then the color of these platform is a little different so the developers should define what design want to appear on what platform they used.

4. Criteria

4.1. Typography design

Color

There are three common structures for a good color. These are triadic, compound and analogous. Choosing color in the same area of color spectrum is called an analogous color scheme. The colors are usually different by their contrast and vibrancy when compared with each other.

Figure 4.1.1

The basic color of the application is used moderate blue color that is based on a selection of colors from same area. The background color is using with the mixture of moderate blue and white. The color of the navigation bar is using very soft blue that can contrast with background color. The background for text in the whole application is mostly used with white. Using the black text color on the white content background makes clear and easy to read. This is the best match and simple color for contrast. Buttons color is used in dark blue to be match with background moderate blue and button text is white color. Crowdsourcing application is showing with a large amount of data so using ordinary white and black text color will not complicate to design and the light blue color represents the nature of facial clinic.

Font

The choosing font is one of the important processes in designing application. Pairing two or more fonts in an application or design is more attractive than using one font style. There are many ways of pairing and this crowdsourcing application will be used the type

of contrast pairing. Contrast between fonts usually provides a winning combination. Any font style can be paired and sometimes the different fonts style will contrast.
The application will be considered using the contrast between sans-serif and serif fonts. This is a simple and classic way of font pairing. Using decorative serif for heading/ titles and, sure-footed sans-serif for body make the design great. So, I planned to use Georgia, the types of serif and Ariel, the types of sans-serif for desktop application.

Size and Spacing

In generally, designers are set their text body at least 12px of font size. However, a larger size of font like 14px is even better to choose for readability. It is easy to decide the body text font size but difficult for headings. The size of text in the application is 12px and around 20px for heading. Too large or small text cannot attract user attention. So, body text size12px is a pretty good one.

Spacing between the lines of text is very helpful tool for readability. Bad spacing can destroy the attraction of text and good spacing makes oppositely. For large amount of blocks, 1.5times line spacing is the best and clear.

Layout

The use of grid is the most important things in making design. Use appropriate scale of grid with the system and it should be formal, usable and looking good.

The main page of application design is designed layout with totally three grids and the left two grids are merged.

4.2. User Group

This application is for desktop used and mainly aim to use in Facial Clinics. So the target users who use the system are generally Facial Doctors. Facial doctors may not expert in using computer system and so the design should not be too complex. Simple design that can easy access for all types of user. Among the user group, not only facial doctors, also assistant doctors and nurse from the clinics are target to use the system.
Actually, the application has a server side user or users who can entry the web data, but in this time, only targets to design for common users.

5. Designing Prototype

5.1. Prototype phase 3

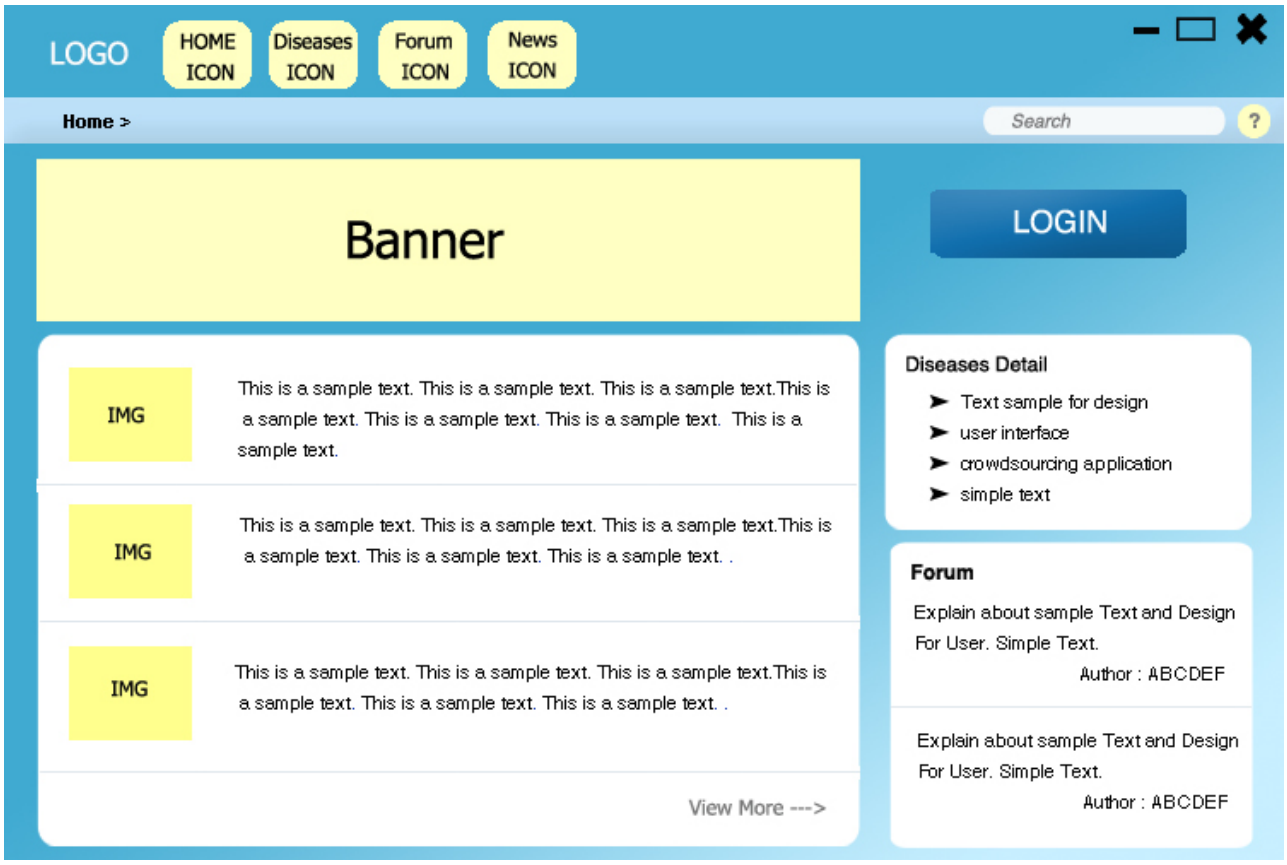


Figure 5.1.1

(See in appendix)

6. Evaluation

6.1. Evaluation Technique

The current applications are very complex and high-level products, usability evaluation can helpfully determine the success or not. Before and after the application deployment, the product usability should be evaluated.

Heuristic Evaluation is an evaluation technique that can evaluate user interface with quick and easy methods. As Heuristic Evaluation requires a few resources, which are money, time or expertise, it is famous in web development. Any developer can use the technique and Heuristic Evaluation is categorized with small parts of test scenarios that use screen shots and paper style, which can simply change from one situation to another.

Firstly, identify the tasks to test that are critical with its objective. Provide evaluators who evaluate the task with goals of the system. Evaluators can break down the goal into relevant tasks and evaluate each of them.

Choosing evaluators are the important part of evaluation technique. The developer can find evaluators who are specialists in software and have knowledge about how this software is applied; they will typically discover 81% to 90% of usability issues from interface design. If the developer can't find specialist evaluators, find people who are novice in software and don't have knowledge about software. They will discover 22% 29% of design usability issues. The Evaluators is to go through the user interface design several times, test and measure the efficiency of this design using the ten heuristic principles.

Analyze task for evaluation

The following are the task for whole system using Hierarchical Task Analysis and I have planned to produce and evaluate prototype for 1 to 4.

1) Access general function
2) Sign up to enter the system with user name
    a. Entry required data
    b. Login to system
3) Access Disease List

    a. Access Disease Detail
4) Access Forum
    a. Write article
    b. Post the article
5) Access News of diseases
6) Access User Info
7) Access Help and manual

Heuristic Evaluation

Number	**Broad Heuristic**
1	Ease of Visible function
2	Use Simple and Natural Language
3	Be Consistent
4	Prevent Error
5	Provide Clearly Marked Exits
6	Minimize User's Memory Load
7	Effective Menu/Command Structure
8	Use Aesthetic and Minimalist Design
9	Deal with Errors in a Positive Manner
10	Provide Help

6.2. Evaluation with evidence

1. Access general function

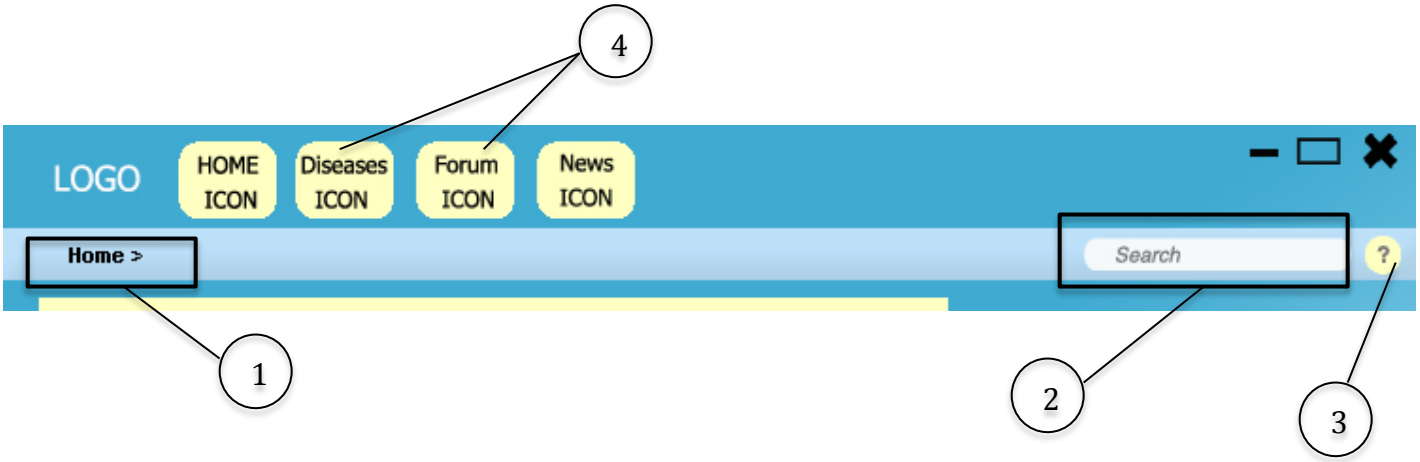
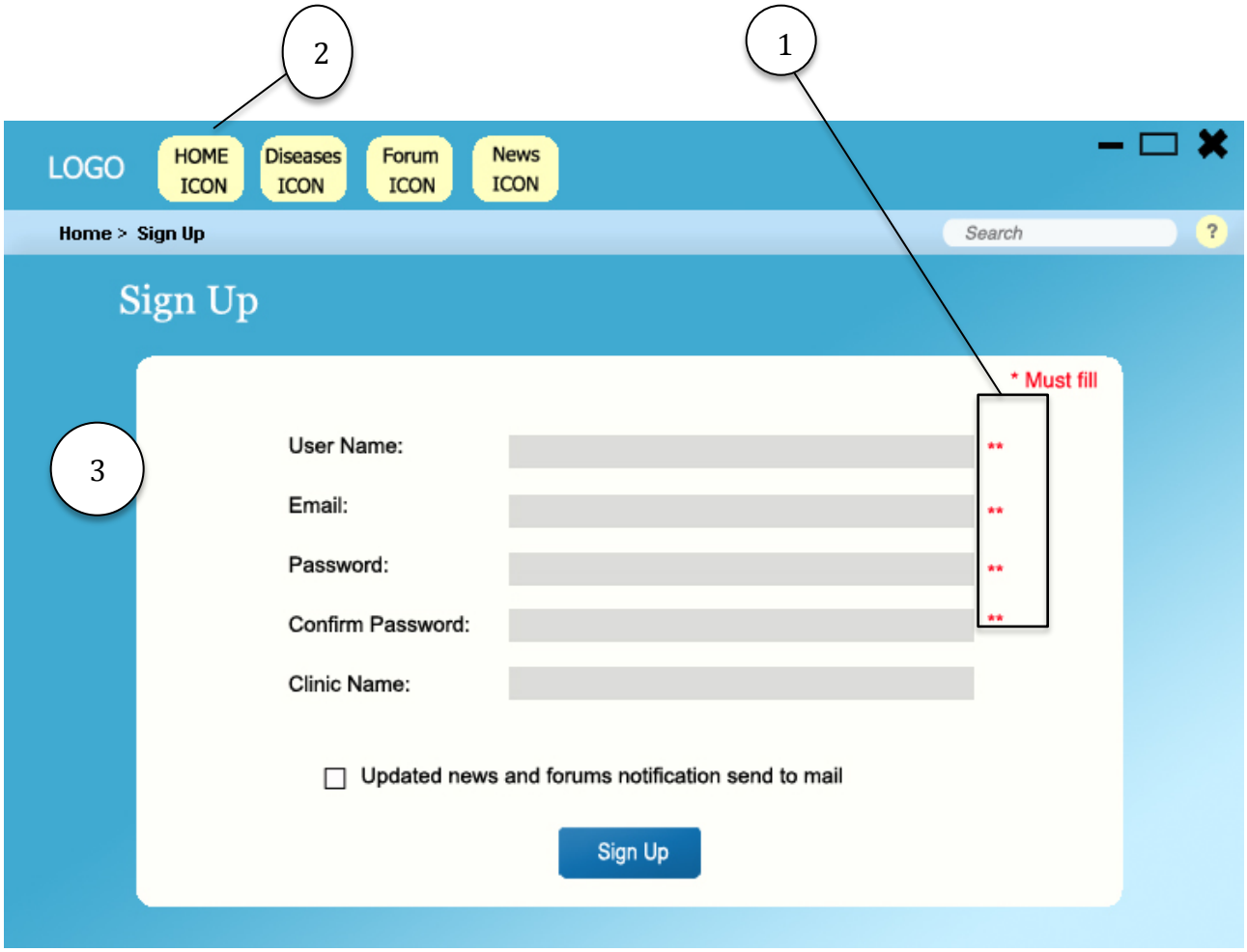


Figure 6.2.1

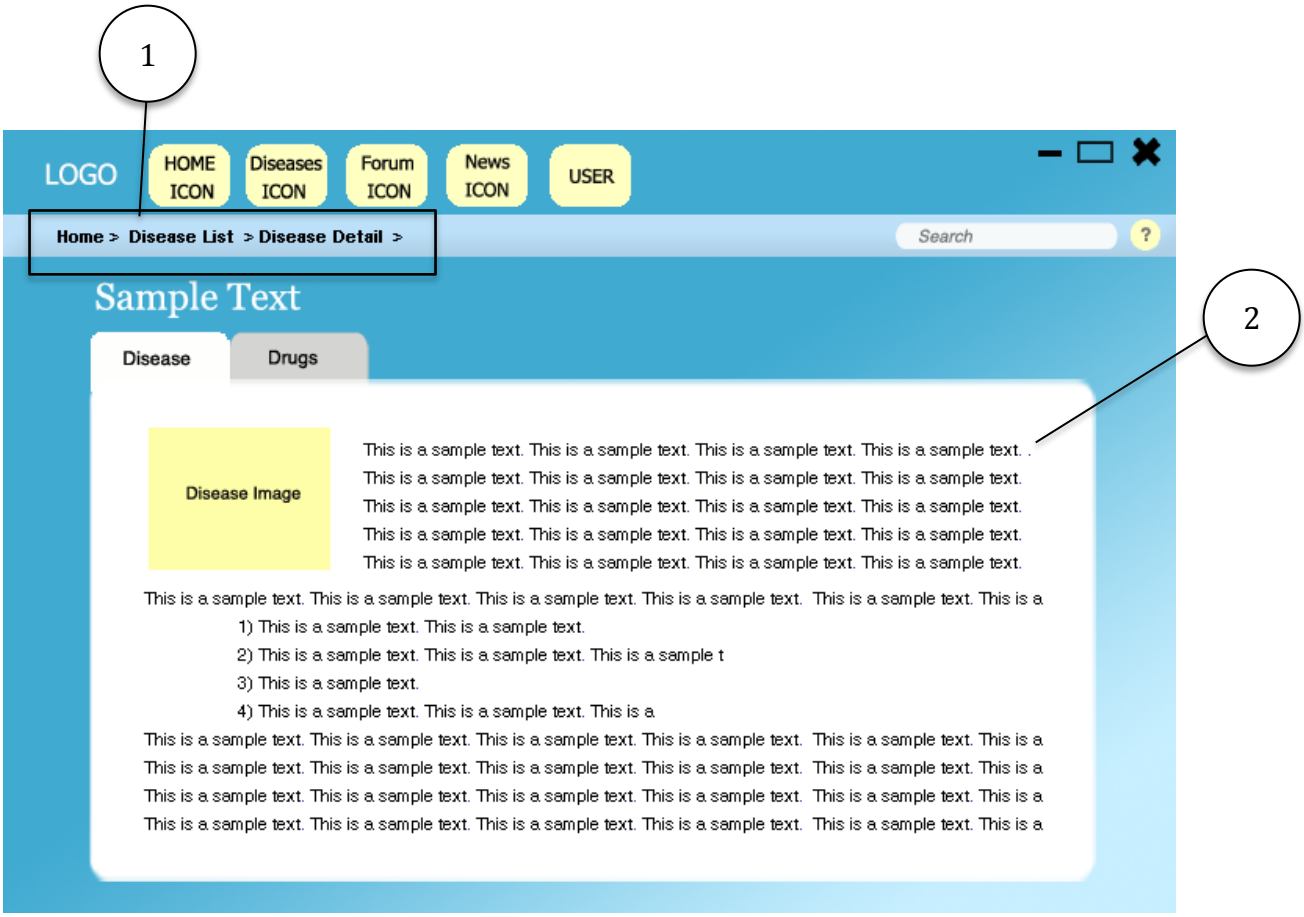
No.	Description	Heuristic number	Evaluation
1	Navigation Bar	#1	Navigation bar helps to indicate where user is exist now
2	Search box	#1	Search box can help to search data easily
3	Help button	#10	Consists of user manual and documentation
4	Icon	#1	Large icon bar is attractive and visible but only icon may be difficult to understand. Should display text under the icon or on mouse over.

1. Sign up to enter the system with user name



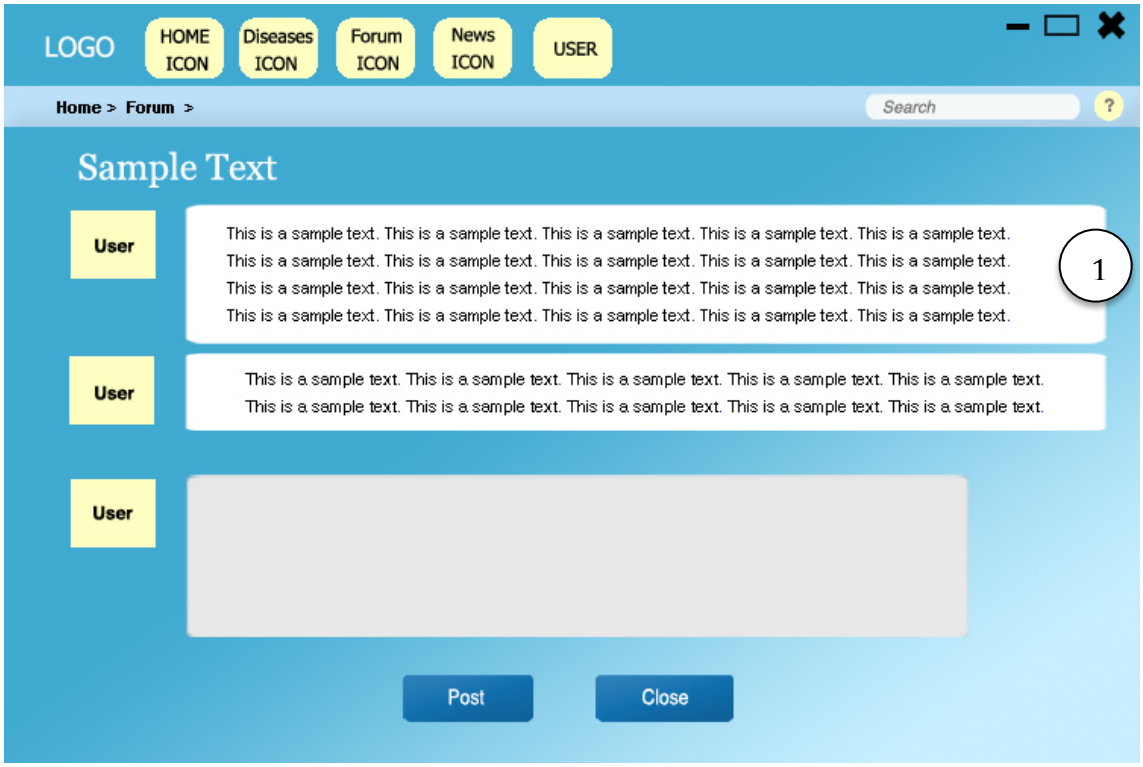
No.	Description	Heuristic number	Evaluation
1	Home Icon	#5	Easily return to home wherever the user exist
2	Error warning	#4	Before user makes errors, prevent error with warning
3	text	#2	Use simple word of language

3.1. Access Disease Detail



No.	Description	Heuristic number	Evaluation
1	Navigation Bar	#1	Navigation bar helps to indicate where user is exist now
2	Page Style	#3	The same type of each page design is also the same and consistent

4. Access Forum



No.	Description	Heuristic number	Evaluation
1	Page Style	#3	The same type of each page design is also the same and consistent

7. Appendix

7.1. Prototype Phase 1

The system is about web-based crowdsourcing application and includes the news and information of facial diseases. There are four pages of prototype for this case study. The main page, Disease detail page, Forum and sign up page. The main page includes all updated news (online necessary) of facial diseases or clinic, forum made by doctors, and the detail of various facial diseases. The diseases detail pages contains the explanation and solution about diseases. Forum contains the various discussions concerned with diseases info, clinic condition or patients. To participate in forum, the user must be member of the system. So, user should fill entry form in sign in page.

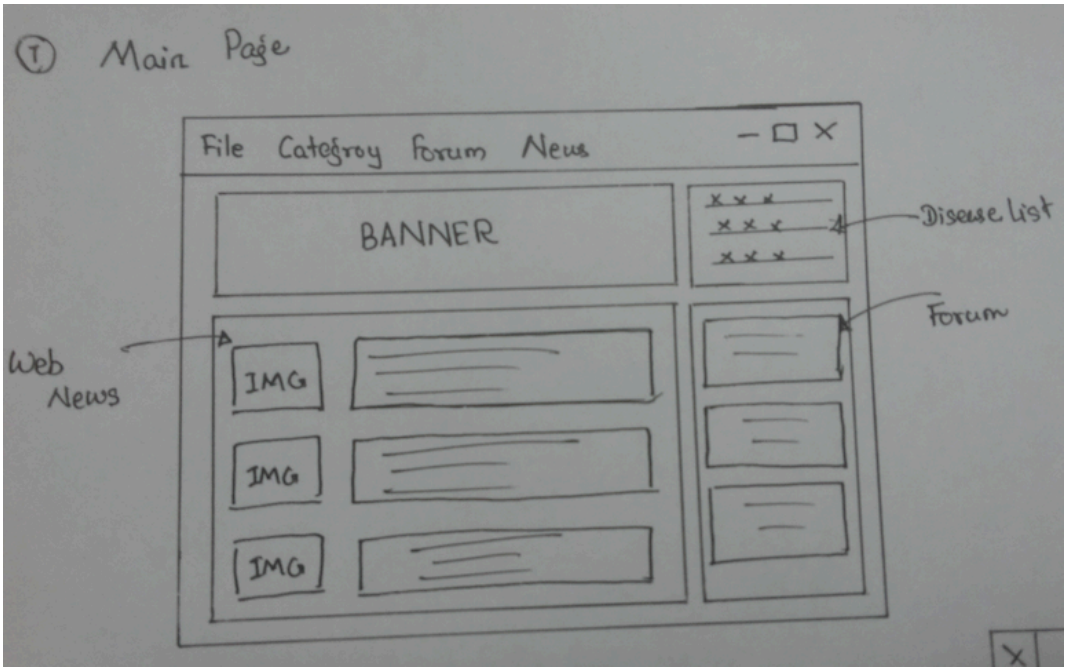


Figure 7.1.1

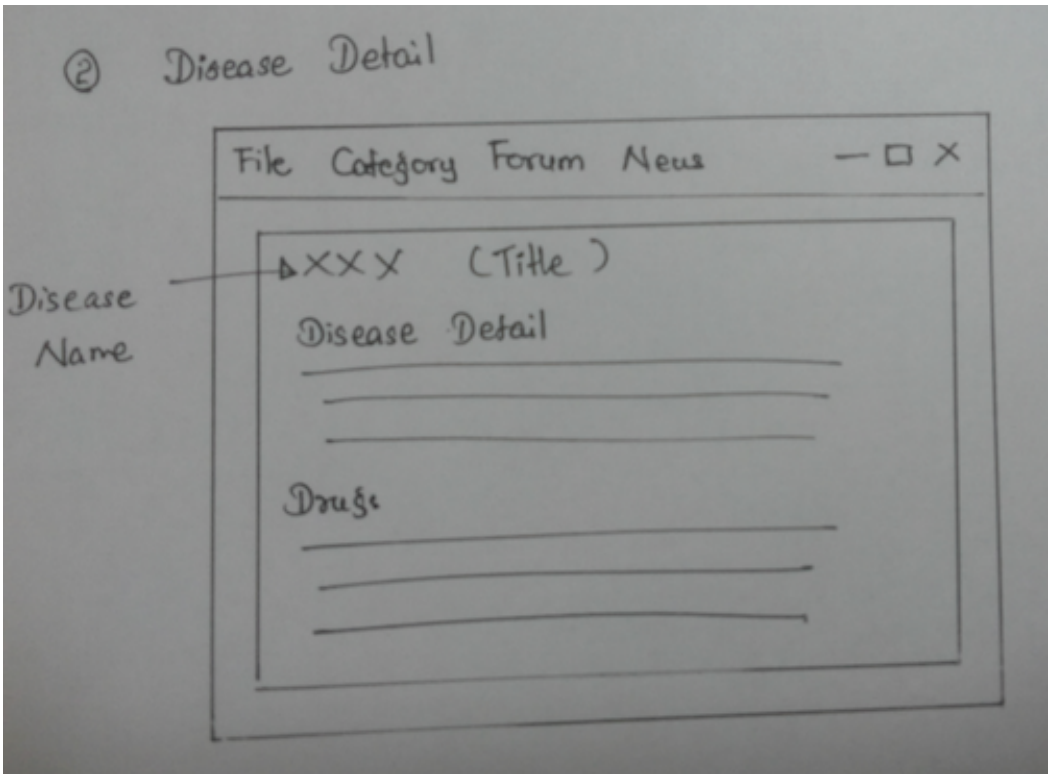


Figure 7.1.2

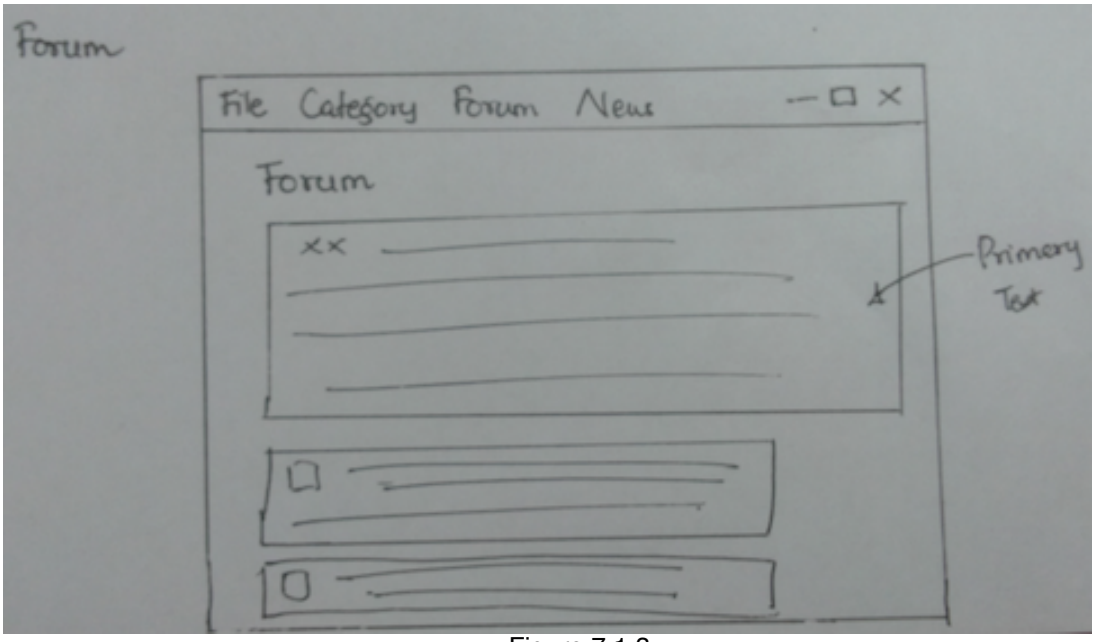


Figure 7.1.3

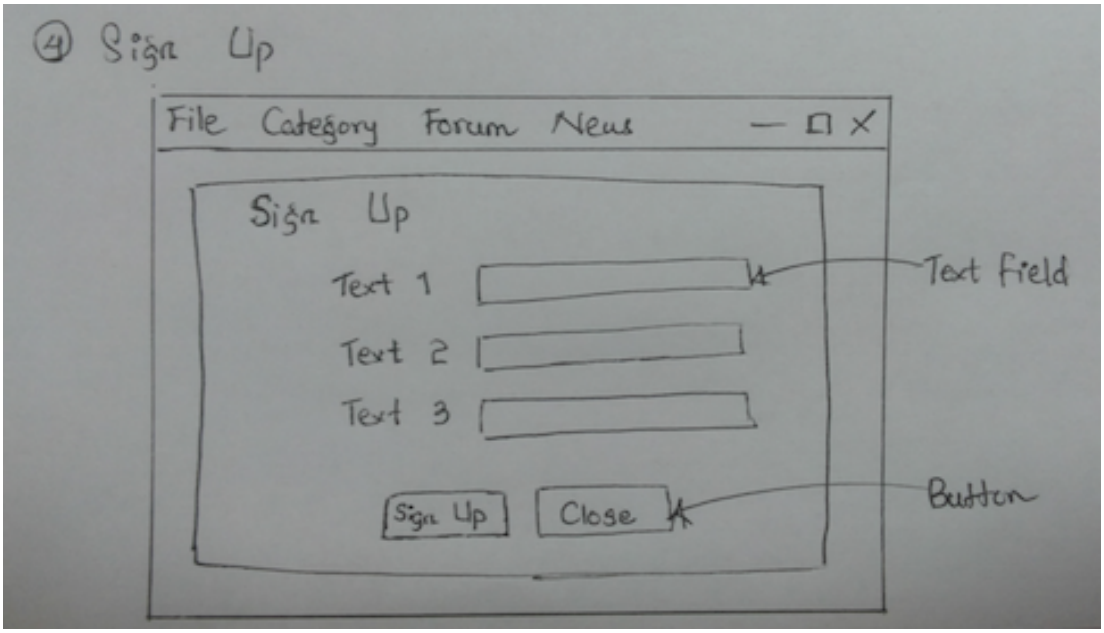


Figure 7.1.4

8.2. Prototype Phase 2

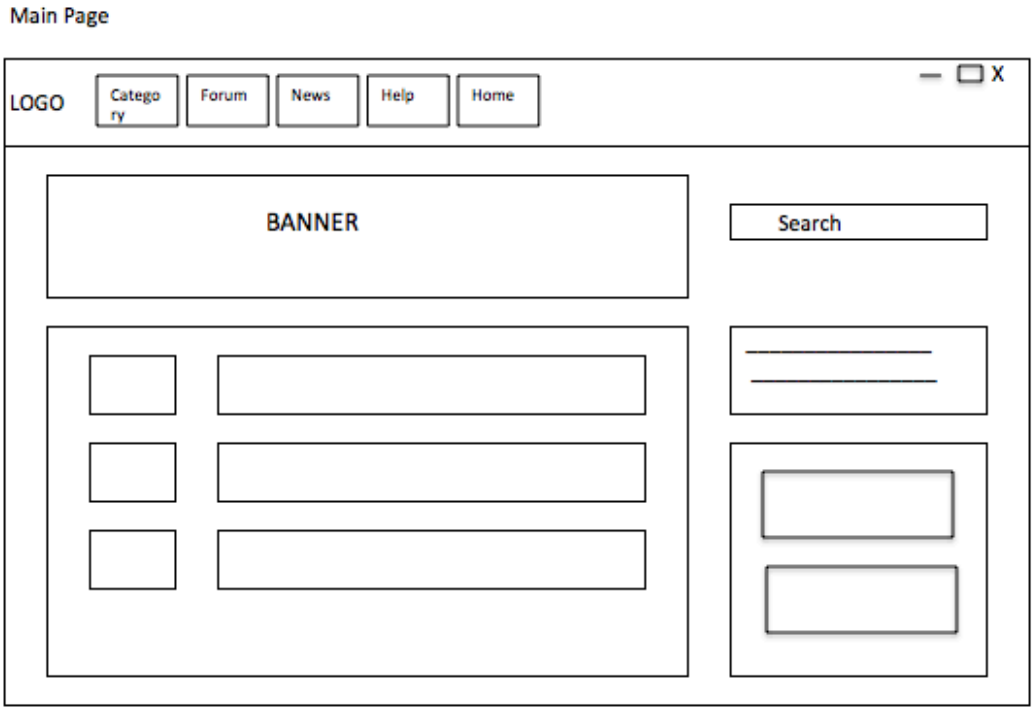


Figure 7.2.1

Disease Detail

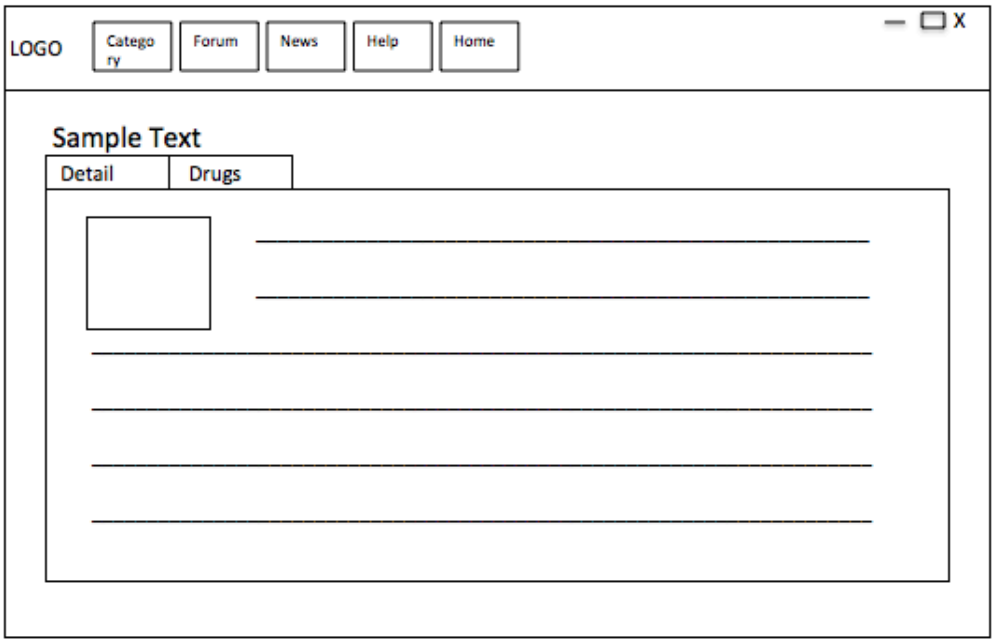


Figure 7.2.2

Forum

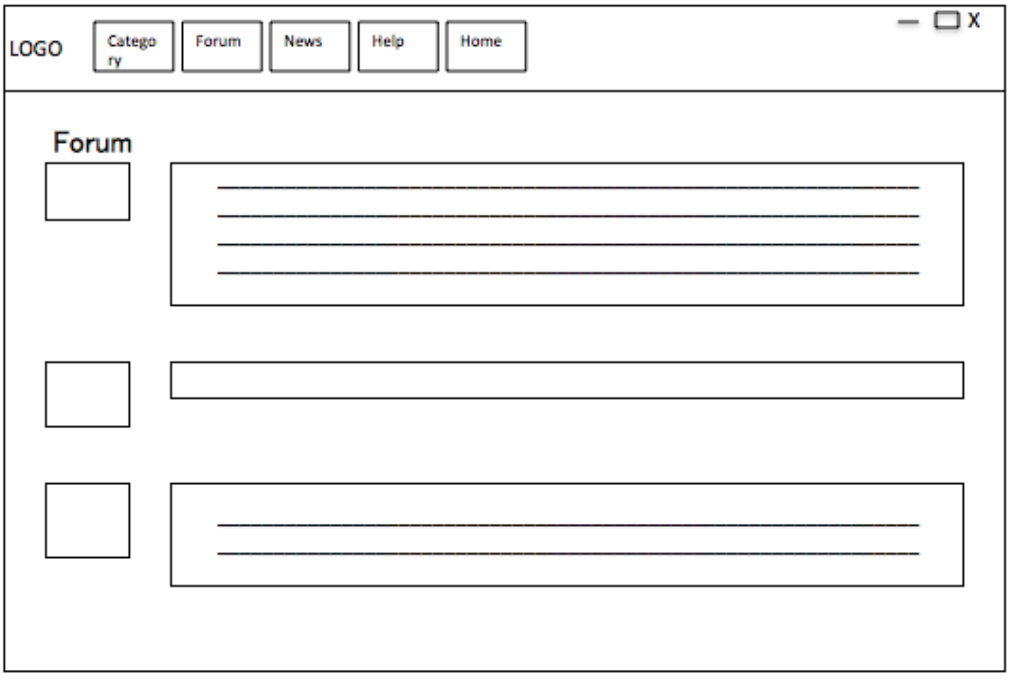


Figure 7.2.3

Sign Up

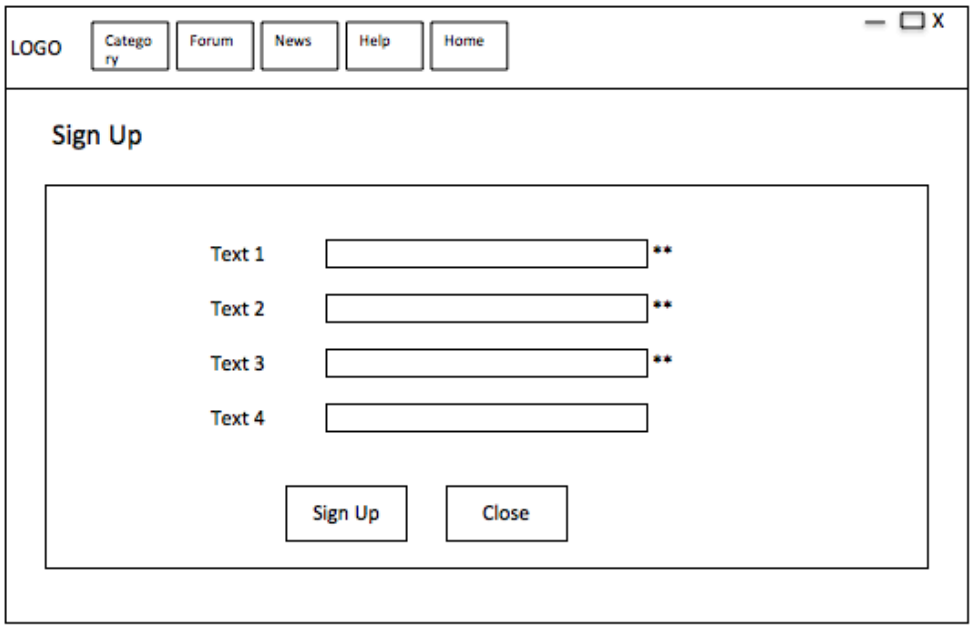


Figure 7.2.4

8.3. Prototype Stage 3

Disease Detail

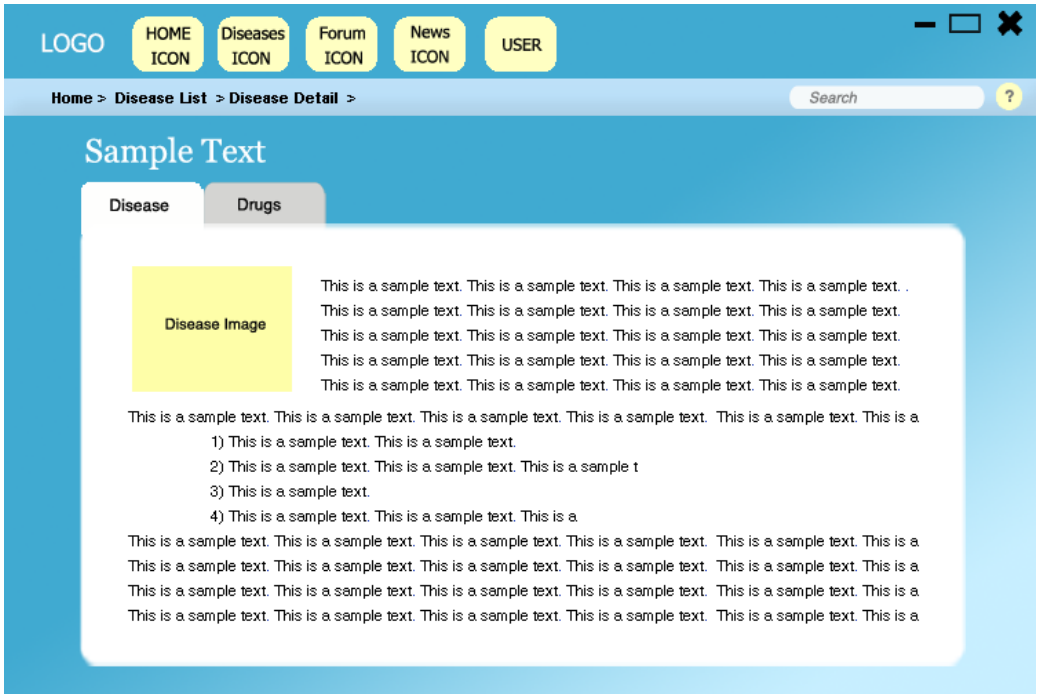


Figure 7.3.1

Forum

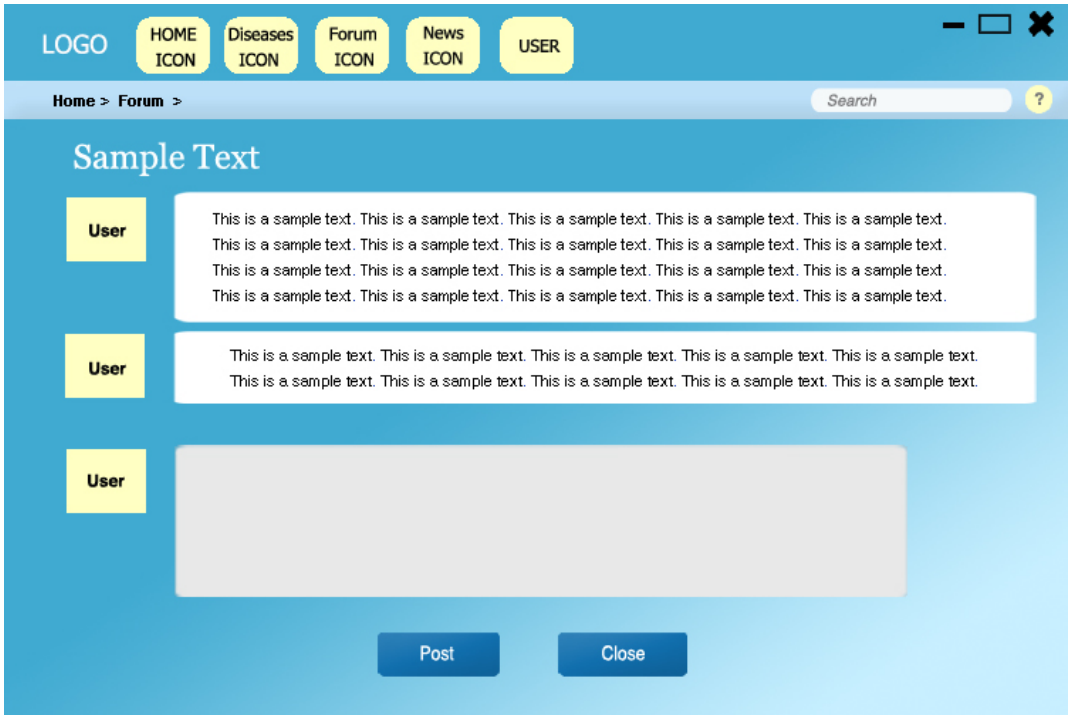


Figure 7.3.2

Sign Up

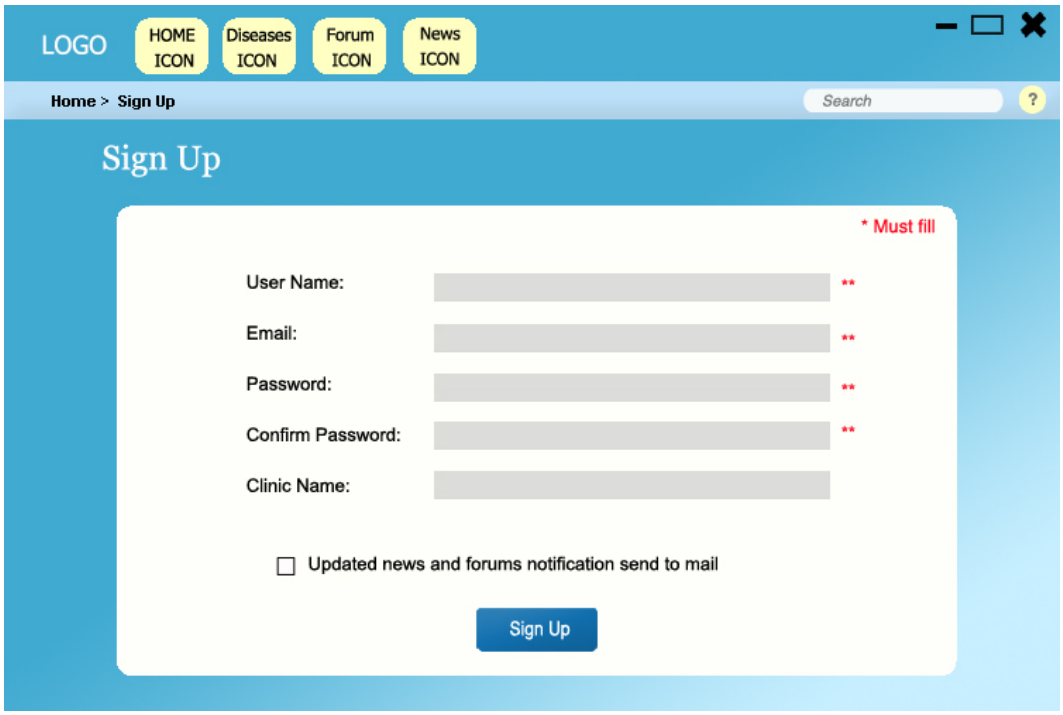


Figure 7.3.3

9. References

1) JAKOB NIELSEN (1995), Usability Heuristics [online] available from: http://www.nngroup.com/articles/ten-usability-heuristics/ [Accessed on 3 November 2014]
2) JAKOB NIELSEN (2008), Application design mistakes [online] available from: http://www.nngroup.com/articles/top-10-application-design-mistakes/ [Accessed on 5 November 2014]
3) JAKOB NIELSEN (2005), Scrolling [online] available from: http://www.nngroup.com/articles/scrolling-and-scrollbars/ [Accessed on 2 November 2014]
4) MSDN(n.d), Error message [online] available from: http://msdn.microsoft.com/en-us/library/windows/desktop/dn742471(v=vs.85).aspx [Accessed on 1 November 2014]
5) Thomas Cannon (2012), Color Theory [online] available from: http://webdesign.tutsplus.com/articles/an-introduction-to-color-theory-for-web-designers--webdesign-1437 [Accessed on 5 November 2014]
6) Max Luzuriaga (2011), Typography [online] available from: http://webdesign.tutsplus.com/articles/choosing-the-right-font-a-practical-guide-to-typography-on-the-web--webdesign-15 [Accessed on 5 November 2014]
7) Lan Yates (2012), Paring Font [online] available from: http://webdesign.tutsplus.com/articles/a-beginners-guide-to-pairing-fonts--webdesign-5706 [Accessed on 5 November 2014]
8) Nicky Danino (2001), Evaluation guide [online] available from: http://www.sitepoint.com/heuristic-evaluation-guide/ [Accessed on 7 November 2014]
9) Ron Garrett and Jennifer Lewis (n.d), Ethical Issues in developing software [online] available from: https://www.scribd.com/doc/10880744/Ethical-Issues-in-Software-Development [Accessed on 9 November 2014]
10) Usability Professional Association (2010), Cognitive Task Analysis [online] available from: http://www.usabilitybok.org/cognitive-task-analysis/ [Accessed on 8 November 2014]