

**Usman Institute of Technology**

**Department of Computer Science**

**Fall 2022**

**CS411- Human Computer Interaction**

**Lab #13**

**Objective:**

Understanding the Work on Evaluation

|  |  |
| --- | --- |
| Name of Student |  |
| Student ID |  |
| Date of Lab Conducted |  |
| Marks Obtained |  |
| Remarks |  |
| Signature |  |

**Usman Institute of Technology**

**Department of Computer Science**

**CS411- Human Computer Interaction**

**Fall 2022**

1. **Objective**

Understanding the Work on Evaluation

**2. Description:**

## **Evaluating designs:**

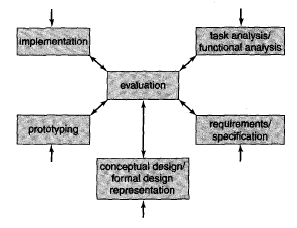
Evaluation is the process of determining the usability and acceptability of the product or design that is measured in terms of a variety of criteria including the number of errors users make using it, how appealing it is, how well it matches the requirements, and so on. Interaction design requires a high level of user involvement throughout development, and this enhances the chances of an acceptable product being delivered.

In most design situations you will find a number of activities concerned with quality assurance and testing to make sure that the final product is "fit-for-purpose."

## **Lifecycle models in HCI**

Another of the traditions from which interaction design has emerged is the field of HCI (human computer interaction). Fewer lifecycle models have arisen from this field than from software engineering and, as you would expect, they have a stronger tradition of user focus. We describe two of these here. The first one, the Star, was derived from empirical work on understanding how designers tackled HCI design problems. This represents a very flexible process with evaluation at its core. In contrast, the second one, the usability engineering lifecycle, shows a more structured approach and hails from the usability engineering tradition.

## **The Star Lifecycle Model**

About the same time that those involved in software engineering were looking for alternatives to the waterfall lifecycle, so too were people involved in HCI looking for alternative ways to support the design of interfaces. In 1989, the Star lifecycle model was proposed by Hartson and Hix (1989) (see Figure 6.13). This emerged from some empirical work they did looking at how interface designers went about their work. They identified two different modes of activity: analytic mode and synthetic mode. The former is characterized by such notions as top-down, organizing, judicial, and formal, working from the systems view towards the user's view; the latter is characterized by such notions as bottom-up, free-thinking, creative and ad hoc, working from the user's view towards the systems view. Interface designers move from one mode to another when designing. A similar behavior has been observed in software designers (Guindon, 1990).

## **Evaluation Techniques**

Evaluation

* tests usability and functionality of system
* occurs in laboratory, field and/or in collaboration with users
* evaluates both design and implementation
* should be considered at all stages in the design life Cycle

## **Evaluating Designs**

* Cognitive Walkthrough
* Heuristic Evaluation

https://sites.google.com/a/nu.edu.pk/hci-060129/lectures-1/heuristic-evaluation

1. **How to Submit**

* Submit lab work in a single pdf/docx/fig file on MS Team/Google Classroom.
* Submit the work as per format given in this manual (No other format will be accepted)
* Lab work (Exercises) file name should be saved with your roll number and course code (e.g. 19B-001-CS\_CSxxx\_LWxx.pdf where CSxxx is course code and LWxx is Lab number)

1. **Exercise(s)**

Considering your project as a sample, you are required to create and conduct the evaluation.

***Cognitive Walkthrough***is a usability evaluation method that is used to identify usability problems in a user interface by simulating the user's thought process as they complete a task. The method involves having a usability expert (or team) walk through a set of tasks using the interface, while thinking out loud and describing the thoughts and actions they would take to complete the task.

The main goal of a cognitive walkthrough is to identify usability problems that may be encountered by actual users, and to identify ways to improve the user interface to make it more user-friendly.

The process of a cognitive walkthrough typically includes the following steps:

**Define the tasks**: Identify the tasks that users will need to complete using the interface, and create a set of scenarios that describe how users will complete these tasks.

Register

Request for registering officer into the system.

Login

Registering user.

Searching information

Correct vehicle number to be entered.

Reporting bugs

Bugs can be informed to the developer via email mode.

Accessing calendar on search vehicle page

Dates should be clearly defined via drop down.

Defining owner details

User have clear understanding for text box instruction.

**Identify the user's goals**: Identify the user's goals for each task, and the actions they will take to complete the task.

Register

Admin associate roles for officer then the user will be available to access the system.

Login

Only registered users can access the system.

Searching information

Rfids can be registered first with the number plate.

Reporting bugs

Correct fields have to be entered.

Accessing calendar on search vehicle page

Dates should be clearly defined via drop down. If dates are not defined correctly the system give error.

Defining owner details

User enter information correctly and important information.

**Walk through the task**: Have a usability expert or team walk through each task and scenario, thinking out loud and describing the thoughts and actions they would take to complete the task.

Register

Admin associate roles for officer then the user will be available to access the system.

Separate GUI for admin for registering different users and all the tasks of officer.

Also maintaining registered officers via this GUI.

Login

Only registered users can access the system.

At first user have to register via requesting from admin.

Searching information

Rfids can be registered first with the number plate.

Incorrect format of number plate can result in error or dialog box.

Also car registration process have to be done first.

Reporting bugs

Correct fields have to be entered.

Bugs have file upload options in which you can upload error screenshot or define that bug.

Accessing calendar on search vehicle page

Dates should be clearly defined via drop down. If dates are not defined correctly the system will give an error.

Defining owner details

User enter information correctly and important information.

Define the important fields with asterisk key for defining important fields without that information the form will not be submitted.

–     Is the effect of the action the same as the user’s goal at that point?

Yes

–     Will users see that the action is available?

Yes

–     Once users have found the correct action, will they know it is the one they need?

Guidance should be added for user. As this is a simple prototype it is providing enough information.

–     After the action is taken, will user understand the feedback they get?

***Heuristic Evaluation*** is a usability evaluation method that is used to identify usability problems in a user interface by having a usability expert (or team) evaluate the interface against a set of established usability principles, known as heuristics.

Heuristics are general guidelines that have been found to be effective in identifying usability problems in user interfaces. They are not hard and fast rules, but rather are based on the expert's experience and knowledge of human-computer interaction.

The process of a heuristic evaluation typically includes the following steps:

**Define the scope of the evaluation**: Identify the areas of the interface that will be evaluated, and the tasks that users will need to complete using the interface.

**Identify the heuristics**: Identify a set of heuristics that will be used to evaluate the interface. Common heuristics include visibility of system status, match between system and the real world, user control and freedom, consistency and standards, error prevention, and recognition rather than recall.

* system behaviour is predictable

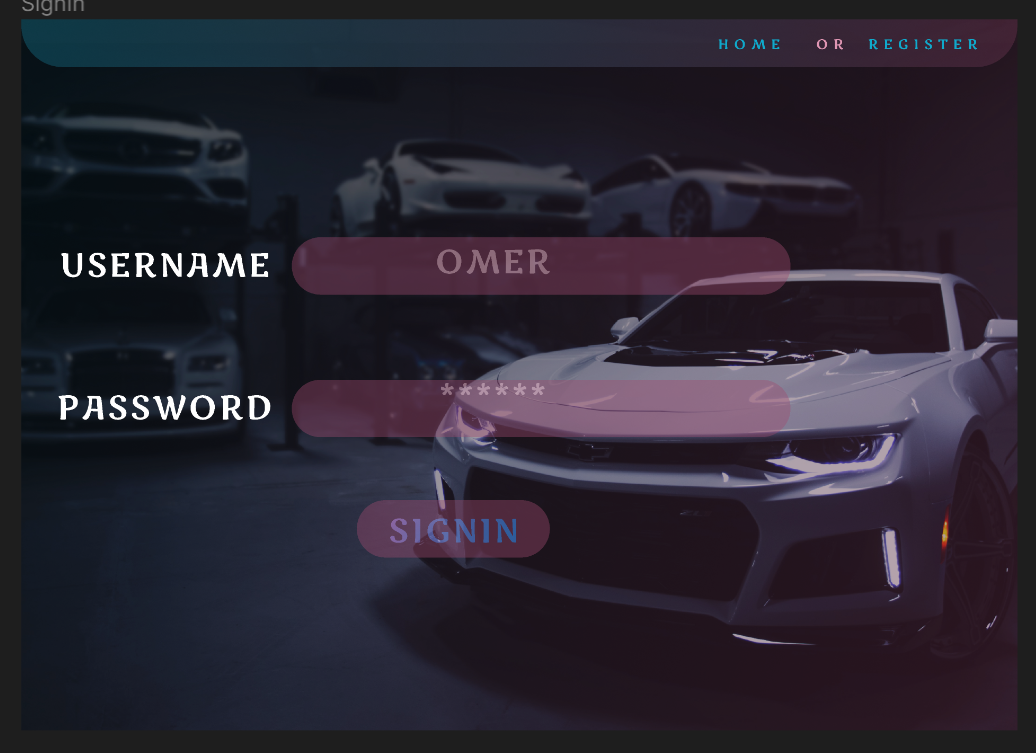
yes

* system behaviour is consistent

yes

* feedback is provided

no

 Chart, waterfall chart

Description automatically generated

Graphical user interface, text

Description automatically generated Graphical user interface, application

Description automatically generated