**HARAMAYA UNIVERSITY**

**COLLEGE OF COMPUTING AND INFORMATICS**

**DEPARTMENT OF COMPUTER SCIENCE**

**OBJECT ORIENTED SOFTWARE DEVELOPMENT**

**DEVELOPING PROTOTYPE:**

**INTELLIGENT TOURIST PROJECT SYSTEM**

**PREPARED BY:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Id** | **Email** |
| **Abrahim Abdo** | **768/13** | [**abre2789@gmail.com**](mailto:abre2789@gmail.com) |
| **Zemzem Mohammed** | **776/13** | [**Zemzemmohammed143@gmail.com**](file:///D:\Data\Master\Software%20Engineering\propasal\Zemzemmohammed143@gmail.com) |
| **Murad Amen** | **772/13** | [**Muradamen10@gmail.com**](file:///D:\Data\Master\Software%20Engineering\propasal\Muradamen10@gmail.com) |
| **Ketama Diriba** | **771/13** | [**Ketama.will@gmail.com**](file:///D:\Data\Master\Software%20Engineering\propasal\Ketama.will@gmail.com) |

**Submitted to: Dr. Fekade. G**

November 5, 2021

Haramaya University, Ethiopia

Table of Contents

[1. Introduction 3](#_Toc86986337)

[1.1 Use Scenario Development 3](#_Toc86986338)

[2. User Interface Design Process 3](#_Toc86986339)

[2.1 Interface Structure Design 4](#_Toc86986340)

[2.2 Interface Design Prototyping 4](#_Toc86986341)

[2.2.1 HTML Prototype 5](#_Toc86986342)

[3. Interface Evaluation 5](#_Toc86986343)

[3.1 Interactive Evaluation 5](#_Toc86986344)

[3.1.1 Formal Usability Testing 6](#_Toc86986345)

# Introduction

A use scenario is an outline of the steps that the users perform to accomplish some part of their work. A use scenario is one commonly used path through a use case. Recall that use cases and data flow diagrams may include multiple ways in which the response to the event can be completed. In our proposed system we used html prototype and interactive approach for evaluation. Then first we start from the scenario in our system during analysis. The system model can be described in terms of its structure and its dynamic interoperation. For Developing prototype the framework is WYSIWYG website software.

## Use Scenario Development

We identify five actors, one for each type of user in the problem statement (tourist, admin, agent. tourism officer and bank manager). As the core functionality of the system is to manage and get tourism service, we first develop scenario, Managetourismservice to elicit and explore this functionality in more detail. By first focusing on a narrow vertical slice of the system, we understand better the client’s expectation of the system, including the boundary of the system and the kinds of interactions between the user and the system. Based on the answers from the client, we refine the scenario accordingly**.**

* **Manage tourism service** scenarios
* **Get report generated** scenarios
* **Access tour package** scenarios
* **Managed tourism service schedule** scenarios

High-level scenarios identified for ITPS. Clients and developers initially briefly describe scenarios. They refine them further to clarify ambiguities or uncover disagreements

# User Interface Design Process

User interface design5 is a five-step process that is iterative—analysts often move back and forth between steps rather than proceed sequentially from step 1 to step 5.

1. First, the analysts examine the DFDs and use cases developed in the analysis phase and interview users to develop use scenarios that describe users’ commonly employed patterns of actions so that the  
   interface can enable users to quickly and smoothly perform these scenarios.
2. Second, the analysts develop the interface structure diagram (ISD) that defines the basic structure of the interface. This diagram (or set of diagrams) shows all the interfaces (e.g., screens, forms, and reports) in the system and how they are connected.
3. Third, the analysts design interface standards, which are the basic design elements on which interfaces in the system are based.
4. Fourth, the analysts create an interface design prototype for each of the individual interfaces in the system, such as navigation controls, input screens, output screens, forms (including preprinted paper forms), and reports.
5. Finally, the individual interfaces are subjected to interface evaluation to determine whether they are satisfactory and how they can be improved.

Interface evaluations almost always identify improvements, so the interface design process is repeated in a cyclical process until no new improvements are identified. In practice, most analysts interact closely with the users during the interface design process, so that users have many chances to see the interface as it evolves rather than waiting for one overall interface evaluation at the end of the interface design process.

## Interface Structure Design

The interface structure design defines the basic components of the interface and how they work together to provide functionality to users. An interface structure diagram (ISD) is used to show how all the screens, forms, and reports used by the system are related and how the user moves from one to another. Most systems have several ISDs, one for each major part of the system. [1]

## Interface Design Prototyping

An interface design prototype is a mock-up or a simulation of a computer screen, form, or report. A prototype is prepared for each interface in the system to show the users and the programmers how the system will perform. In the “old days,” an interface design prototype was usually specified on a paper form that showed what would be displayed on each part of the screen. Paper forms are still used today, but more and more interface design prototypes are being built with computer tools instead of on paper. The three most common approaches to interface design prototyping are storyboards, HTML prototypes, and language prototypes. [2]

### HTML Prototype

One of the most common types of interface design prototypes used today is the HTML prototype. As the name suggests, an HTML prototype is built with the use of Web pages created in HTML (hypertext mark-up language). The designer uses HTML to create a series of Web pages that show the fundamental parts of the system. The users can interact with the pages by clicking on buttons and entering pretend data into forms (but because there is no system behind the pages, the data are never processed). The pages are linked together so that, as the user clicks on buttons, the requested part of the system appears. HTML prototypes are superior to storyboards in that they enable users to interact with the system and gain a better sense of how to navigate among the different screens. However, HTML has limitations—the screens shown in HTML will never appear exactly like the real screens in the system (unless, of course, the real system will be a Web system in HTML).

# Interface Evaluation

The objective of interface evaluation is to understand how to improve the interface design. Interface design is subjective; there are no formulas that guarantee a great user interface. Most interface designers intentionally or unintentionally design an interface that meets their personal preferences, which may or may not match the preferences of the users. The key message, therefore, is to have as many people as possible evaluate the interface—and the more users, the better. Most experts recommend involving at least 10 potential users in the evaluation process.

As with interface design prototyping, interface evaluation can take many different forms, each with different costs and different levels of detail. Four common approaches are heuristic evaluation, walk-through evaluation, interactive evaluation, and formal usability testing. As with interface design prototyping, the different parts of a system can be evaluated by different techniques. [2]

## **Interactive Evaluation**

With an interactive evaluation, the users themselves actually work with the HTML or language prototype in one-on-one sessions with members of the project team. (An interactive evaluation cannot be used with a storyboard. As the user works with the prototype (often by going through the use scenarios or just navigating at will through the system), he or she tells the project team members what he or she likes and doesn’t like and what additional information or functionality is needed. As the user interacts with the prototype, team members record the situation when the user appears to be unsure of what to do, makes mistakes, or misinterprets the meaning of an interface component. If the pattern of uncertainty, mistakes, or misinterpretations recurs across several evaluation sessions with several users, it is a clear indication that those parts of the interface need improvement.

### Formal Usability Testing

Formal usability testing is commonly done with commercial software products and products developed by large organizations that will be widely used through the organization. As the name suggests, it a very formal— almost scientific—process that can be used only with language prototypes (and systems that have been completely built and are awaiting installation or shipping). As with interactive evaluation, usability testing is done in one-on-one sessions in  
which a user works directly with the software. However, it is typically done in a special lab equipped with video cameras and special software that records each and every keystroke and mouse operation so that they can be replayed to help in understanding exactly what the user did.

|  |  |
| --- | --- |
| Standard matrix | Description |
| Layout | The interface should be a series of areas on the screen that are used consistently for different purposes—for example, a top area for commands and navigation, a middle area for information to be input or output, and a bottom area for status information. |
| Content awareness | Users should always be aware of where they are in the system and what information is being displayed. |
| Aesthetics | Interfaces should be functional and inviting to users through careful use of white space, colors, and fonts. There is often a trade-off between including enough white space to make the interface look pleasing and losing so much space that important information does not fit on the screen. |
| User experience | Although ease of use and ease of learning often lead to similar design decisions, there is sometimes a trade-off between the two. Novice users or infrequent users of software will prefer ease of learning, whereas frequent users will prefer ease of use. |
| Consistency | Consistency in interface design enables users to predict what will happen before they perform a function. It is one of the most important elements in ease of learning, ease of use, and aesthetics. |
| Minimize user effort | The interface should be simple to use. Most designers plan on having no more than three mouse clicks from the starting menu until users perform work |

# References

|  |  |
| --- | --- |
| [1] | Alan Dennis, Barbara Haley Wixom, Roberta M. Roth, Systems analysis and design 5th ed., United States of America: John Wiley & Sons, Inc., 2012. |
| [2] | Alan Dix, Janet E. Finlay, Gregory D. Abowd and Russell Beale, Human–Computer Interaction, Harlow: Prentice-Hall, 2004. |