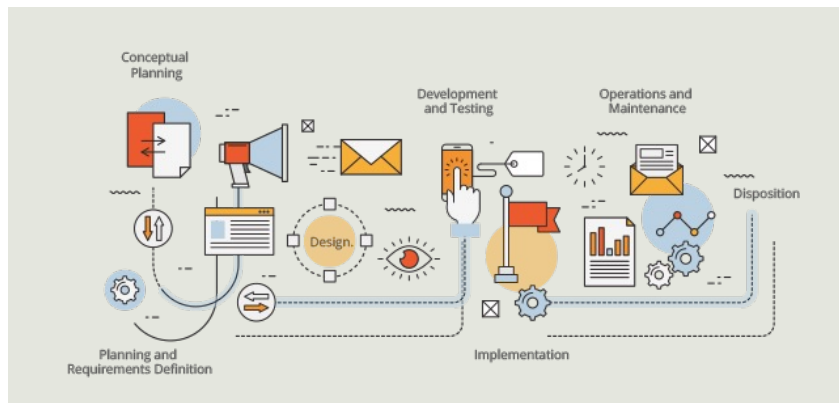


# Software Engineering

## User Interface Design

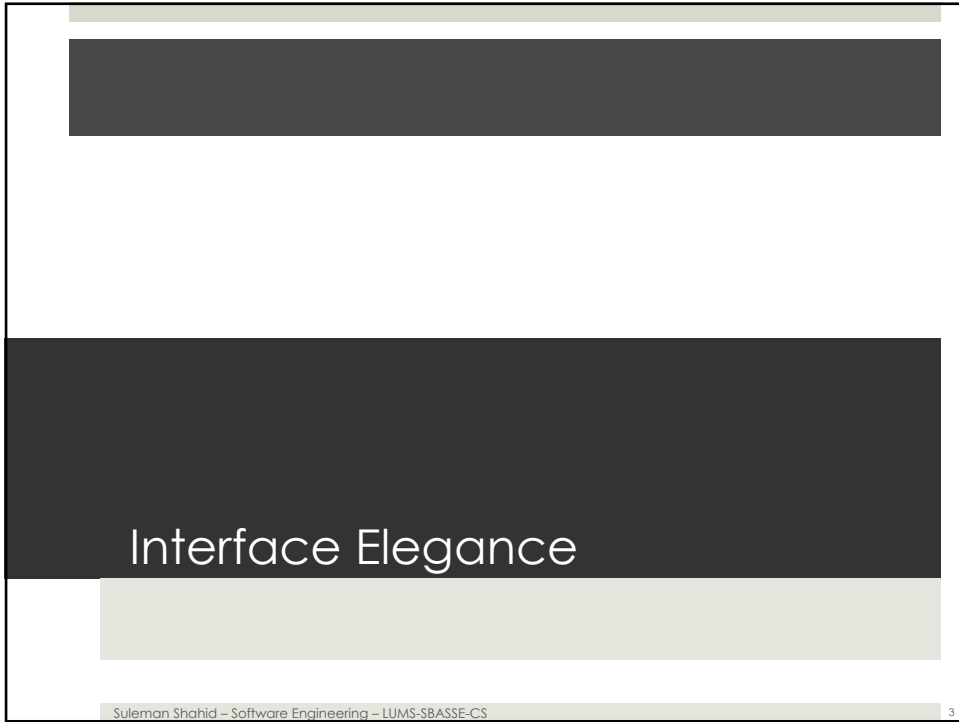
Suleman Shahid



1

# Code Elegance

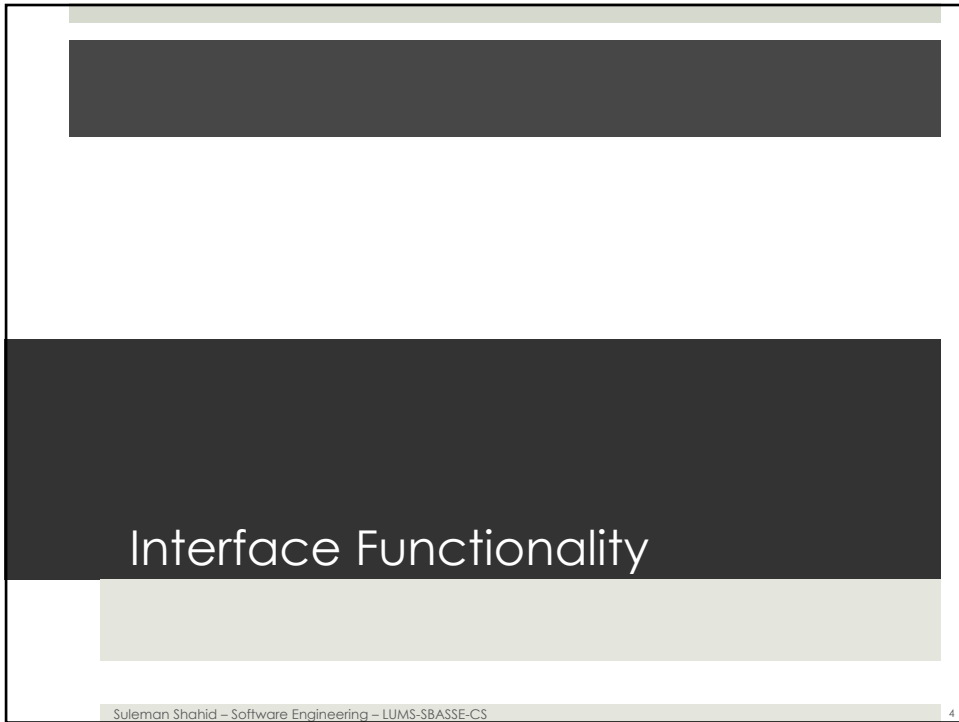
2



A presentation slide with a dark grey header bar at the top. Below the header is a large white rectangular area. At the bottom of the slide is a dark grey footer bar containing the text "Suleman Shahid – Software Engineering – LUMS-SBASSE-CS" and a small number "3" on the right side.

## Interface Elegance

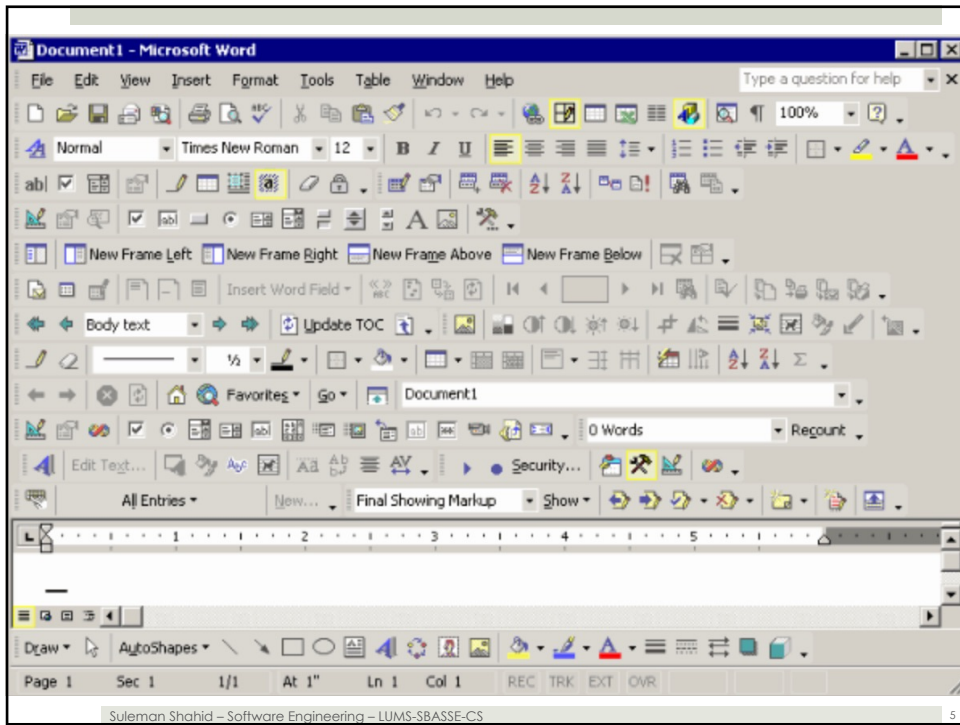
3



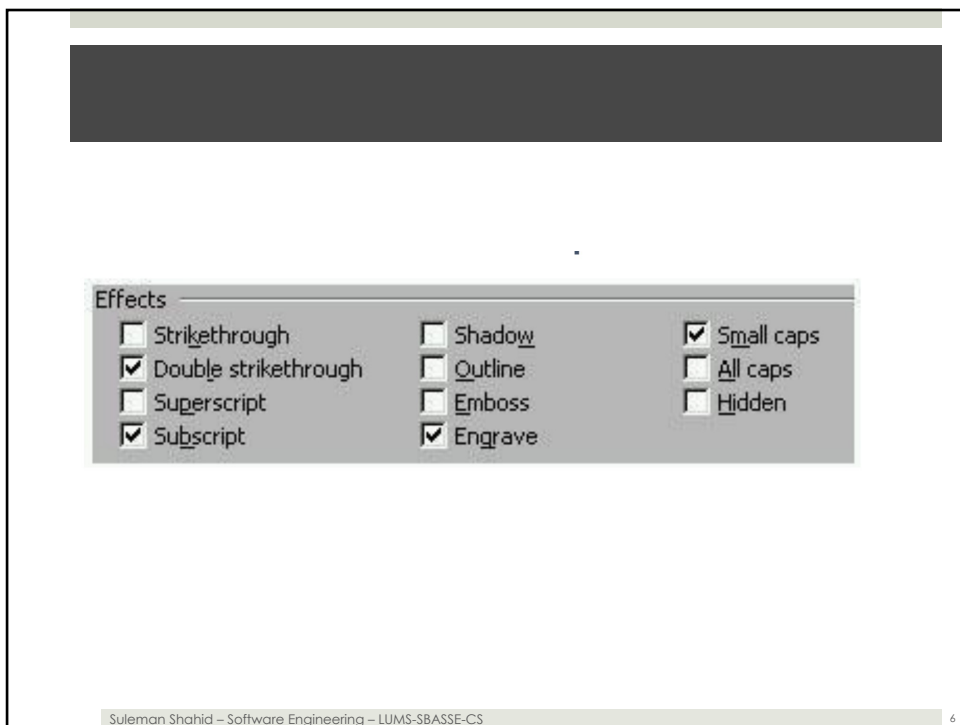
A presentation slide with a dark grey header bar at the top. Below the header is a large white rectangular area. At the bottom of the slide is a dark grey footer bar containing the text "Suleman Shahid – Software Engineering – LUMS-SBASSE-CS" and a small number "4" on the right side.

## Interface Functionality

4



5



6

**Server Error in '/' Application.**

**A transport-level error has occurred when receiving results from the server. (provider: Session Provider, error: 19 - Physical connection is not usable)**

**Description:** An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

**Exception Details:** System.Data.SqlClient.SqlException: A transport-level error has occurred when receiving results from the server. (provider: Session Provider, error: 19 - Physical connection is not usable)

**Source Error:**

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

**Stack Trace:**

```
[SqlException (0x80131904): A transport-level error has occurred when receiving results from the server. (provider: Session Provider, error: 19 - Physical c
System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) +1753346
System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) +5295154
System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHasConnectionLock, Boolean asyncClose) +242
System.Data.SqlClient.TdsParserStateObject.ReadSniFrom(TdsParserStateObject stateObj, UInt32 error) +555
System.Data.SqlClient.TdsParserStateObject.ReadSniSyncOverAsync() +262
System.Data.SqlClient.TdsParserStateObject.TryReadNetworkPacket() +39
System.Data.SqlClient.TdsParserStateObject.TryPrepareBuffer() +64
System.Data.SqlClient.TdsParserStateObject.TryReadByte(Byte& value) +86
System.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler,
System.Data.SqlClient.SqlDataReader.TryConsumeMetaData() +59
System.Data.SqlClient.SqlDataReader.get_MetaData() +90
System.Data.SqlClient.SqlCommand.FinishExecuteReader(SqlDataReader ds, RunBehavior runBehavior, String resetOptionsString) +365
System.Data.SqlClient.SqlCommand.RunExecuteReaderTds(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, Boolean async, Int32 time
System.Data.SqlClient.SqlCommand.RunExecuteReader(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, String method, TaskCompleti
System.Data.SqlClient.SqlCommand.RunExecuteReader(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, String method) +53
System.Data.SqlClient.SqlCommand.ExecuteReader(CommandBehavior behavior, String method) +134
System.Data.SqlClient.SqlCommand.ExecuteReader(CommandBehavior behavior) +41
System.Data.Common.DbCommand.ExecuteReader(CommandBehavior behavior) +10
System.Data.EntityClient.EntityCommandDefinition.ExecuteStoreCommands(EntityCommand entityCommand, CommandBehavior behavior) +437
[EntityCommandExecutionException: An error occurred while executing the command definition. See the inner exception for details.]
System.Data.EntityClient.EntityCommandDefinition.ExecuteStoreCommands(EntityCommand entityCommand, CommandBehavior behavior) +507
System.Data.Objects.Internal.ObjectQueryExecutionPlan.Execute(ObjectContext context, ObjectParameterCollection parameterValues) +730
System.Data.Objects.ObjectQuery`1.GetResults(Nullable`1 forMergeOption) +131
System.Data.Objects.ObjectQuery`1.System.Collections.Generic.IEnumerable<T>.GetEnumerator() +36
System.Linq.Enumerable.FirstOrDefault<IEnumerable`1, Source> +152
System.Data.Objects.EF.Linq.ObjectQueryProvider.<GetElementFunction>b__1(IEnumerable`1 sequence) +41
System.Data.Objects.EF.Linq.ObjectQueryProvider.ExecuteSingle(IEnumerable`1 query, Expression queryRoot) +59
System.Data.Objects.EF.Linq.ObjectQueryProvider.System.Linq.IQueryProvider.Execute(Expression expression) +133
```

Suleman Shahid – Software Engineering – LUMS-SBASSE-CS 7

7

**E-mail**



**Password**

**Error!**

**E-mail**

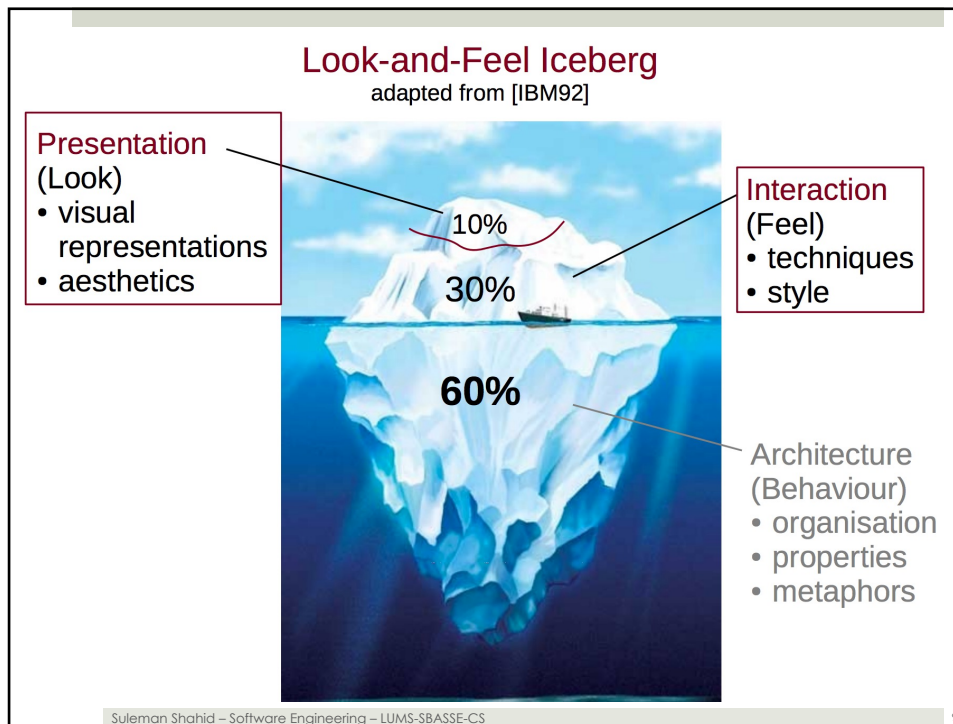
**Password**

Password incorrect. If you don't remember your password, [reset it](#)

Suleman Shahid – Software Engineering – LUMS-SBASSE-CS 8

8



9

## Definitions

- What is Graphical User Interface?

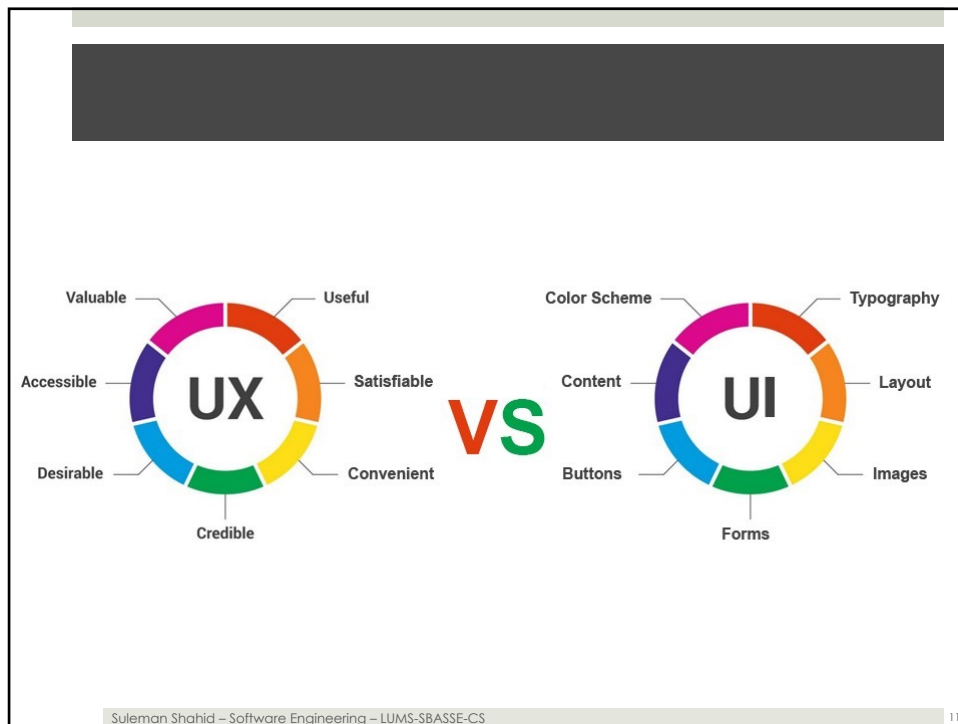
GUI

- What is User Interface?

UI

Suleman Shahid – Software Engineering – LUMS-SBASSE-CS 10

10



11

## Concept of user interface

- User interface is point of connection between user and system
- Soft interface vs. Hard interface
- Soft interface (our focus) handles data inputs and outputs

12 of 19 Suleman Shahid - Software Engineering - LUMS-SBASSE-CS

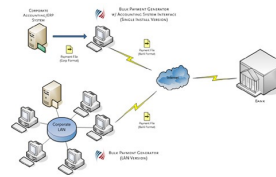
12

## Concept of user interface (cont.)

- User interface requires interaction between user and system to produce inputs and outputs



- In contrast, *system interface* require minimal or no human intervention



13 of 19

Suleman Shahid – Software Engineering – LUMS-SBASSE-CS

13

## User Interface Design Objectives

To design a better user interface, use the following objectives:

- Match the user interface to the task.
- Make the user interface efficient.
- Provide appropriate feedback to users.
- Generate usable queries.
- Improve productivity of knowledge workers.

Suleman Shahid – Software Engineering – LUMS-SBASSE-CS

14-1

14

## Types of User Interfaces

There are several types of user interfaces:

- Natural-language interfaces.
- Question-and-answer interfaces.
- WIMP interface.
  - Graphical User Interfaces (GUIs).
- Gesture based interfaces.
- Voice-based interfaces
- Command-language interfaces.

Suleman Shahid – Software Engineering – LUMS-SBASSE-CS

14-1

15

## Good GUI Rules

1. Understand People
2. Be Careful of Different Perspectives
3. Design for Clarity
4. Design for Consistency
5. Provide Visual Feedback
6. Be Careful With Audible Feedback
7. Keep Text Clear

Suleman Shahid – Software Engineering – LUMS-SBASSE-CS

16

16

## Good GUI Rules

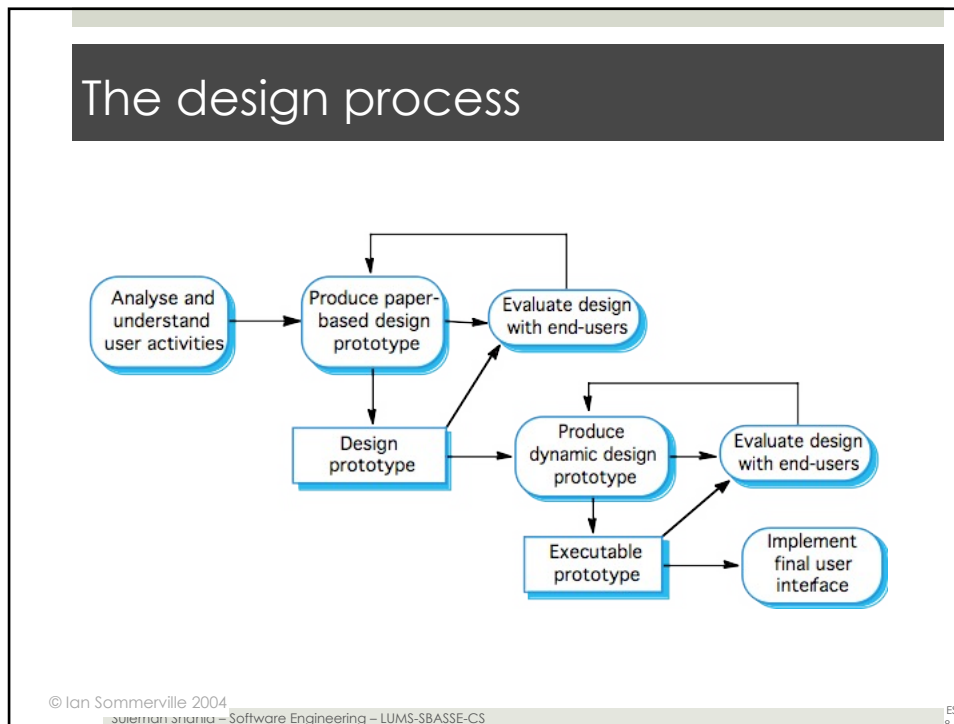
8. Provide Traceable Paths
9. Provide Keyboard Support
10. Watch the Presentation Model
11. Use Modal vs. Modeless Dialogs Appropriately
12. Use Controls Correctly

17

## The UI design process

- UI design is an *iterative process* involving close liaisons between users and designers.
- The 3 core activities in this process are:
  - *User analysis*. Understand what the users will do with the system;
  - *System prototyping*. Develop a series of prototypes for experiment;
  - *Interface evaluation*. Experiment with these prototypes with users.

18



19

## User-centred design

- Sensitise software engineers to key issues underlying the design rather than the implementation of user interfaces
- User-centred design is an approach to UI design where the needs of the user are paramount and where the user is involved in the design process
- UI design always involves the development of prototype interfaces

20

20

## User Interface Design Principles

<b>Principle</b>	<b>Description</b>
<i>User familiarity</i>	Use terms and concepts <i>familiar</i> to the user.
<i>Consistency</i>	Comparable operations should be activated in the <i>same way</i> . Commands and menus should have the same format, etc.
<i>Minimal surprise</i>	If a command operates in a known way, the user should be able to <i>predict</i> the operation of comparable commands.
<i>Feedback</i>	Provide the user with visual and auditory feedback, maintaining <i>two-way communication</i> .

Suleman Shahid – Software Engineering – LUMS-SBASSE-CS

ESE 8.21

21

## User Interface Design Principles (2)

<b>Principle</b>	<b>Description</b>
<i>Memory load</i>	Reduce the amount of information that must be remembered between actions. <i>Minimize</i> the memory load.
<i>Efficiency</i>	Seek efficiency in dialogue, motion and thought. <i>Minimize keystrokes and mouse movements</i> .
<i>Recoverability</i>	Allow users to <i>recover from their errors</i> . Include undo facilities, confirmation of destructive actions, 'soft' deletes, etc.
<i>User guidance</i>	Incorporate some form of <i>context-sensitive user guidance</i> and assistance.

Suleman Shahid – Software Engineering – LUMS-SBASSE-CS

ESE 8.22

22

## Colour Use Guidelines

Colour can help the user **understand complex information structures**.

- Don't use (only) colour to *communicate meaning*!
  - Open to *misinterpretation* (colour-blindness, cultural differences ...)
  - *Design for monochrome then add colour*
- Use colour coding to support user tasks
  - highlight exceptional events
  - allow users to control colour coding
- Use *colour change* to show *status change*
- Don't use *too many* colours
  - Avoid colour pairings *which clash* *which clash*
- Use colour coding *consistently*

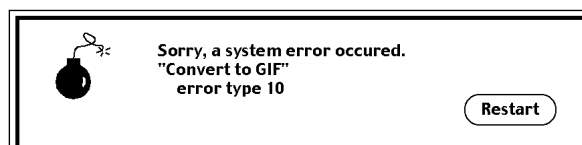
Suleman Shahid – Software Engineering – LUMS-SBASSE-CS

ESE  
3.22

23

## Error Message Guidelines

- *Speak the user's language*
- Give *constructive advice* for recovering from the error
- Indicate *negative consequences* of the error (e.g., possibly corrupted files)
- Give an *audible or visual cue*
- Don't make the user feel guilty!



Suleman Shahid – Software Engineering – LUMS-SBASSE-CS

ESE  
3.22

24

## Design factors in message wording

Context	The user guidance system should be aware of what the user is doing and should adjust the output message to the current context.
Experience	As users become familiar with a system they become irritated by long, 'meaningful' messages. However, beginners find it difficult to understand short terse statements of the problem. The user guidance system should provide both types of message and allow the user to control message conciseness.
Skill level	Messages should be tailored to the user's skills as well as their experience. Messages for the different classes of user may be expressed in different ways depending on the terminology which is familiar to the reader.
Style	Messages should be positive rather than negative. They should use the active rather than the passive mode of address. They should never be insulting or try to be funny.
Culture	Wherever possible, the designer of messages should be familiar with the culture of the country where the system is sold. There are distinct cultural differences between Europe, Asia and America. A suitable message for one culture might be unacceptable in another.

Suleman Shahid – Software Engineering – LUMS-SBASSE-CS

25

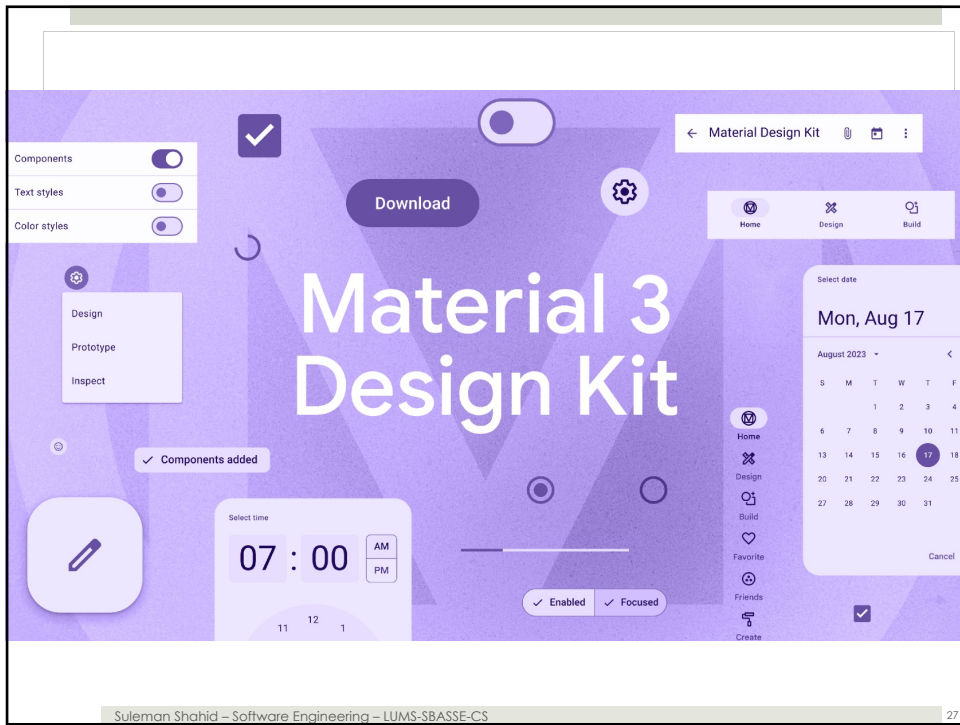
25

## HCI principles

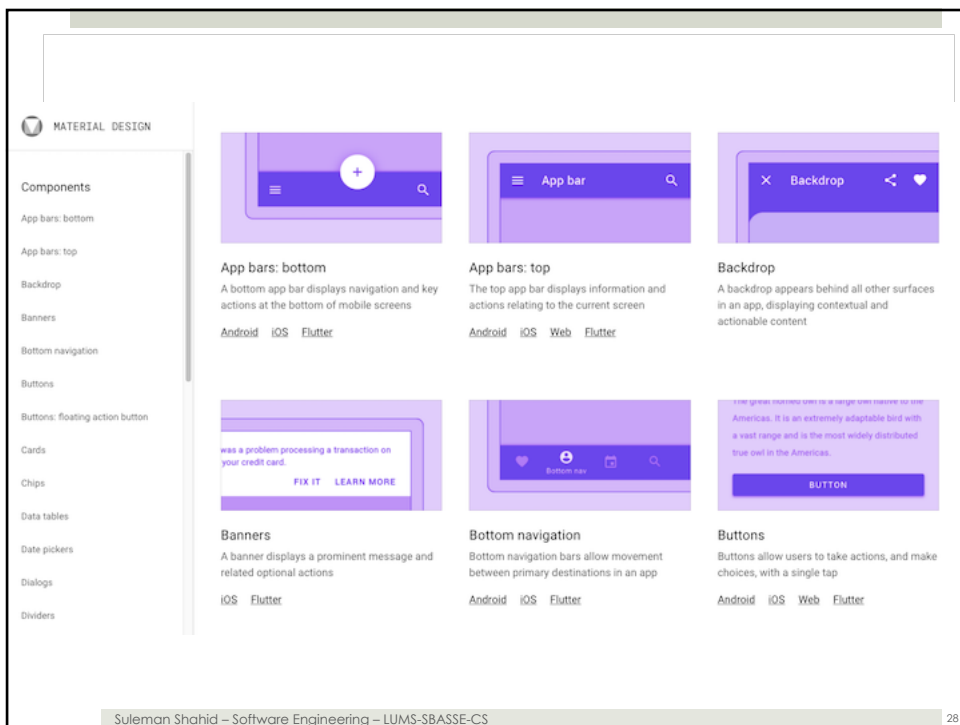
■ Shneiderman's "golden rules" for good design (REF 3C 2S):

- Reversal of user's actions should be simple
- Errors corrected simply
- Feedback should be given to user
- Control to user - not system
- Consistency in screen design
- Closure of dialogues
- Shortcuts offered to users
- Short-term user's memory not to be overloaded

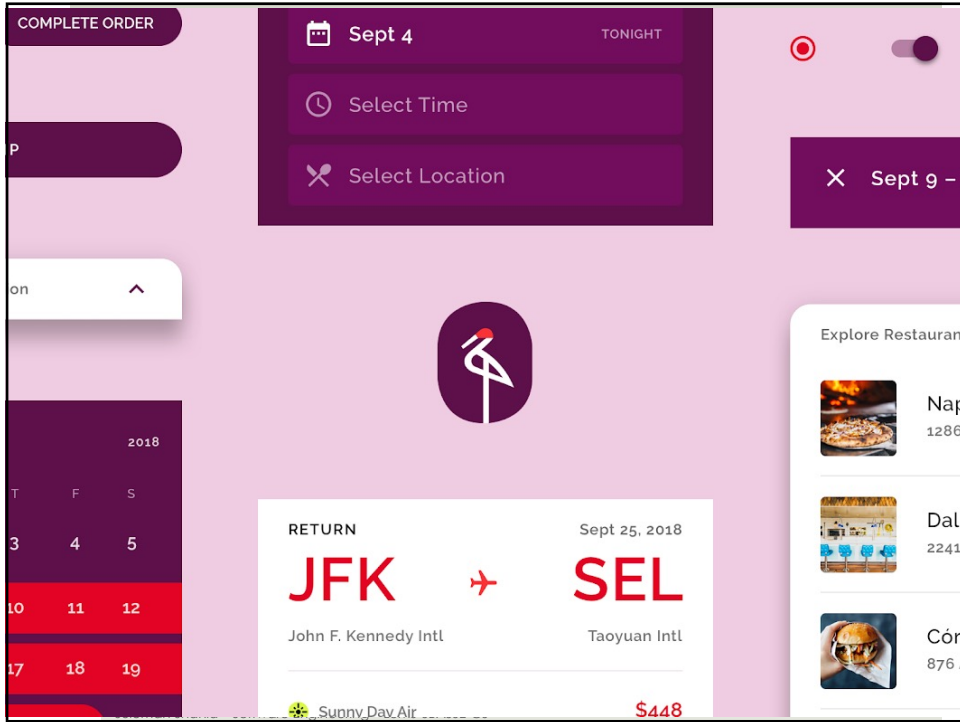
26



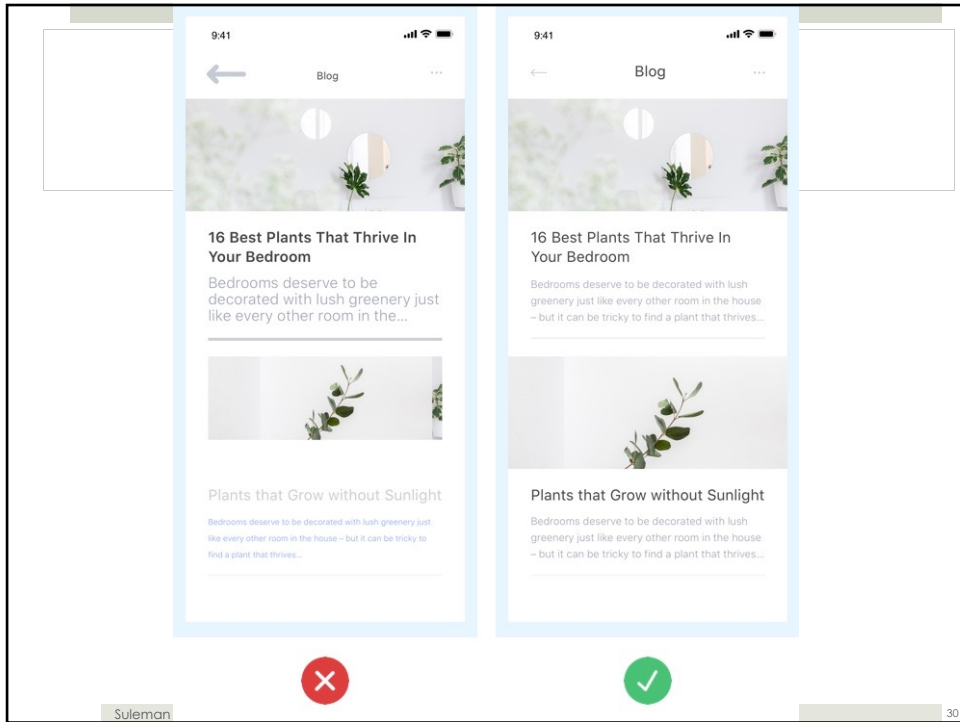
27



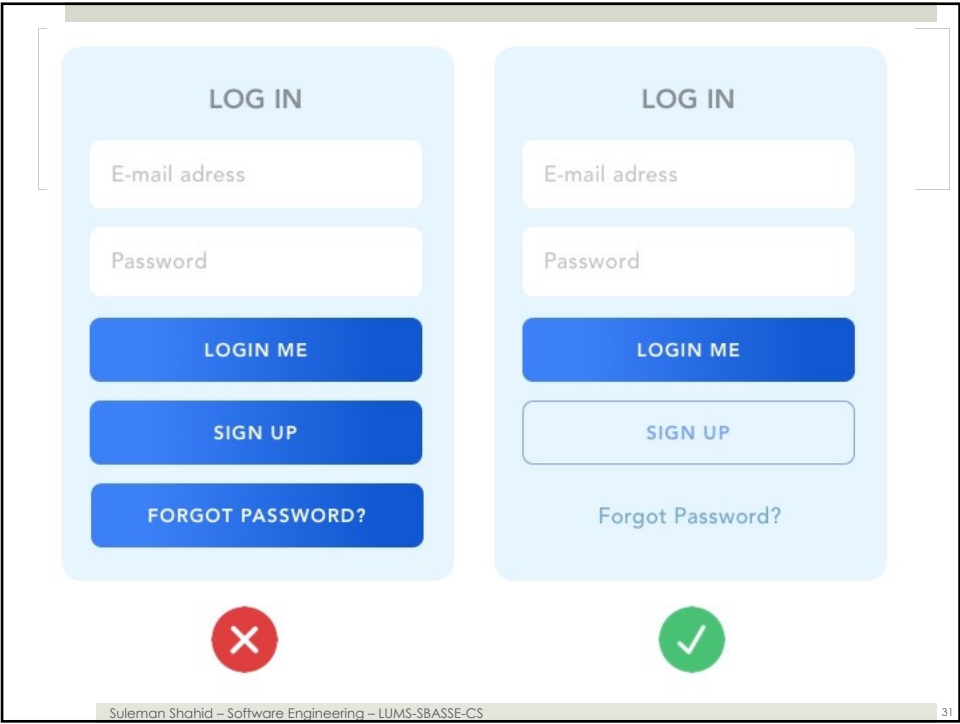
28



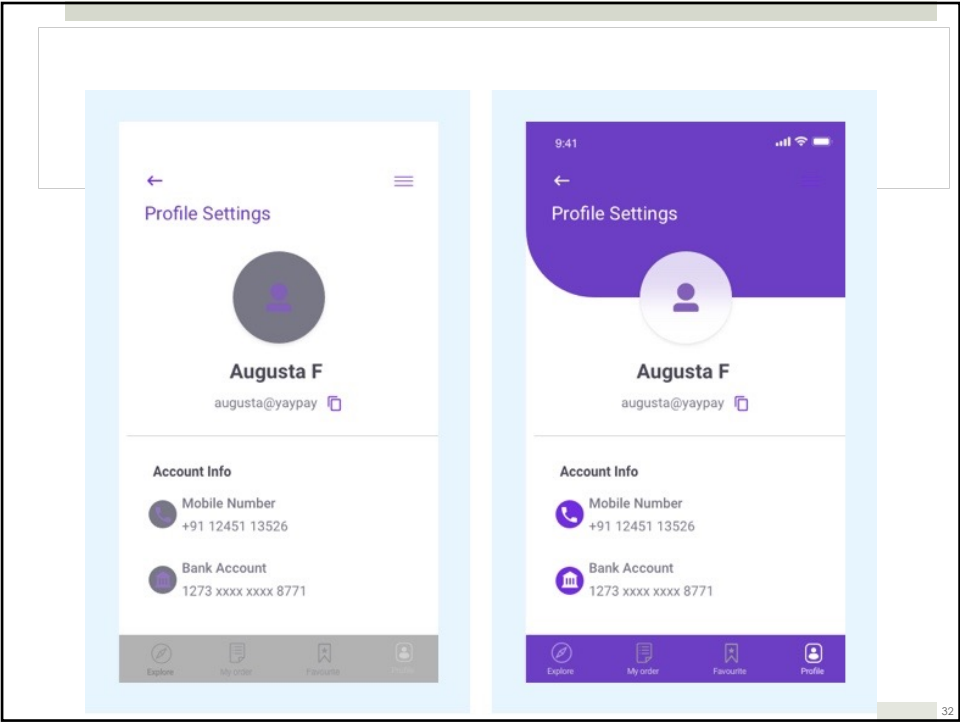
29



30



31



32

Prototyping Tools

Suleman Shahid - Software Engineering - LUMS-SBASSE-CS 33

33

Thanks

Suleman Shahid - Software Engineering - LUMS-SBASSE-CS 34

34