Human Computer Interaction

Module 3- HCI Design

Dr. V. Nivethitha

VIT Chennai

The Overall Design Process

HCI design includes all of the preparatory activities required to develop an interactive software product that will provide a high level of usability and a good user experience.

1. **Requirements analysis:**

- Any software design starts with careful analysis of the functional requirements.
- For interactive software with a focus on the user experience, we take a particular look at functions that are to be activated directly by the user through interaction **(functional-task requirements)**
- functions that are important in realizing certain aspects of the user experience **(functional-UI requirements)** -example is an automatic functional feature of adjusting the display resolution of a streamed video based on the network traffic.
- **nonfunctional UI requirements**, which are UI features (rather than computational functions) that are not directly related to accomplishing the main application task

The Overall Design Process

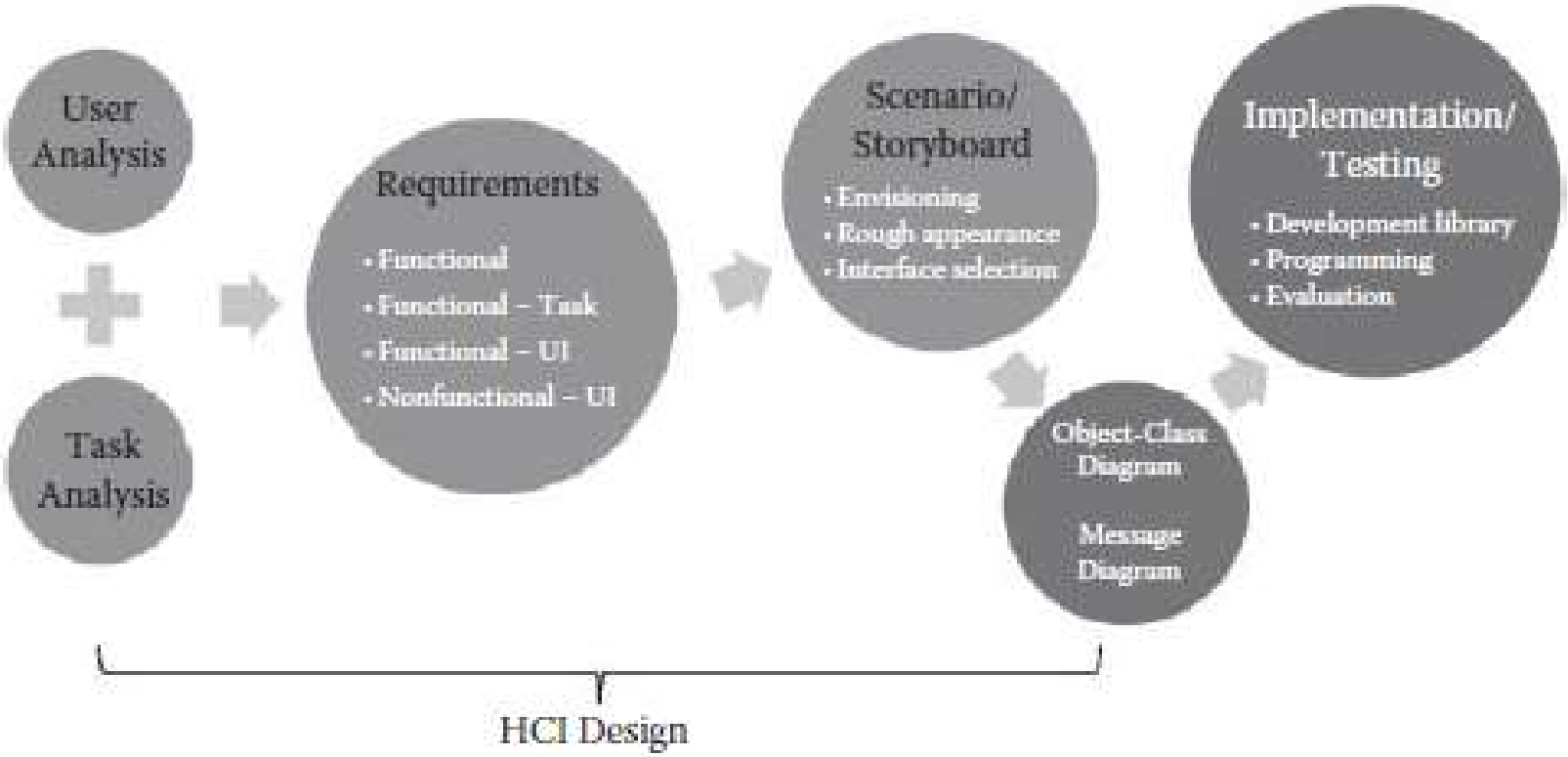


Figure 4.1 The overall iterative HCI design process (as a precursor to implementation).

The Overall Design Process

2. **User analysis:**

- a user analysis is an essential step in HCI design.
- The results of the user analysis will be reflected back to the requirements, and this could identify additional UI requirements (functional or nonfunctional).
- It is simply a process to reinforce the original requirements analysis to further accommodate the potential users in a more complete way.

3. **Scenario and task modeling:**

- identifying the application task structure and the sequential relationships between the different elements.
- With a crude task model, we can also start to draw a more detailed scenario or storyboard to envision how the system would be used and to assess both the appropriateness of the task model and the feasibility of the given requirements.

The Overall Design Process

- Selecting the actual software or hardware interface.
- Object-class diagram, message diagrams, and the use cases for preliminary implementation and programming

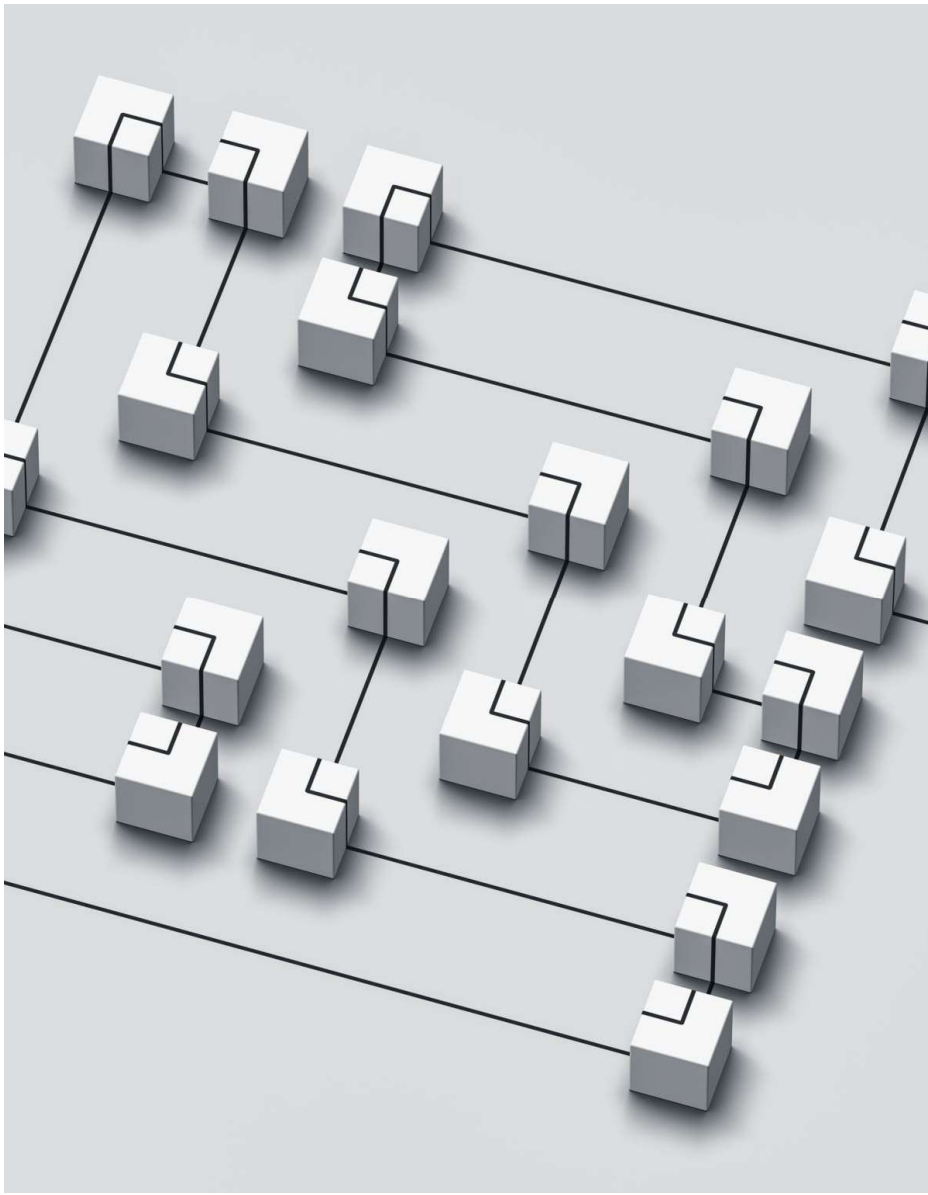
4. Interface selection and consolidation:

- For each of the subtasks and scenes in the storyboard—particularly software interface components (e.g., widgets), interaction technique (e.g., voice recognition), and hardware (sensors, actuators, buttons, display, etc.)—choices will be made.
- The chosen individual interface components need to be consolidated into a practical package

Interface Selection Options

Hardware Platforms

- Different interactions and subtasks may require various individual devices (sensors and displays).
- The choice of a design configuration for the hardware interaction platform is largely determined by the characteristics of the task/application that necessitates a certain operating environment.



Hardware Platforms

- **Desktop (stationary):** Monitor (typical size: 17–42 in.; resolution: 1280 × 1012 or higher); keyboard, mouse, speakers/ headphones (microphone)
- Suited for: Office-related tasks, time-consuming/serious tasks, multitasking
- **Smartphones/handhelds (mobile):** LCD screen (typical size: 3.5–5 in., resolution: 720 × 1280 or higher, weight ≈ 120 g), buttons, touch screen, speaker/headphones, microphone, camera, sensors (acceleration, tilt, light, gyro, proximity, compass, barometer), vibrators, mini “qwerty” keyboard
- Suited for: Simple and short tasks, special-purpose tasks

Hardware Platforms

- **Tablet/pads (mobile):** LCD screen (typical size: 7–10 in., resolution: 720 × 1280 or higher, weight ≈ 700 g), buttons, touch screen, speaker/headphones, microphone, camera, vibrators, sensors (acceleration, tilt, light, gyro, proximity, compass, barometer)
- Suited for: Simple, mobile, and short tasks, but those that require a relatively large screen
- **Embedded (stationary/mobile):** LCD/LED screen (typical size: less than 3–5 in., resolution: low), buttons, special sensors, and output devices (touch screen, speaker, microphone, special sensors); embedded devices may be mobile or stationary and offer only very limited interaction for a few simple functionalities

Hardware Platforms

- ***TV/consoles (stationary)*:** LCD/LED screen (typical size: >42 in., resolution: HD), button-based remote control, speaker,microphone, game controller, special sensors, peripherals (camera, wireless keyboard)
- **Suited for:** TV-centric tasks, limited interaction, tasks that need privacy (e.g., wild-gesture-based games in the living room).
- **Kiosks/installations (stationary):** LCD screen (typical size: 10–13 in., resolution: low to medium), buttons, speaker, touch screen, special sensors, peripherals (microphone, camera, RFID/credit-card reader, heavy-duty keyboard)
- **Suited for:** Public users and installations, limited interaction, short series of selection tasks, monitoring tasks

Software Interface Components

- **Windows/layers:** Modern desktop computer interfaces are designed around windows, which are visual output channels and abstractions for individual computational processes.
- For a single application, a number of subtasks may be needed concurrently and thus must be interfaced through multiple windows.
- One window among the many (or task) would be “active,” and this window becomes “focused” by placing the mouse cursor over it or by an explicit click.

Software Interface Components

- **Icons:** Interactable objects may be visually represented as a compact and small pictogram such as an icon (and similarly as an “earcon” for the aural modality).

- Clickable icons are simple and intuitive. As a compact representation designed for facilitated interaction, icons must be designed to be as informative or distinctive as possible despite their small size and compactness.

Software Interface Components

- **Menus:** Menus allow activations of commands and tasks through selection (recognition) rather than recall.
- Typical menus are organized as a one-dimensional list or a two-dimensional (2-D) array of items (represented in text or as icons/earcons).
- Selection of a menu item involves three subtasks:

  (a) activating the menu and laying out the items (if not already activated by default)

  (b) visually scanning and moving through the items (and scrolling if the display space is not sufficient to contain and show the whole menu of items at once

  (c)choosing the wanted item.

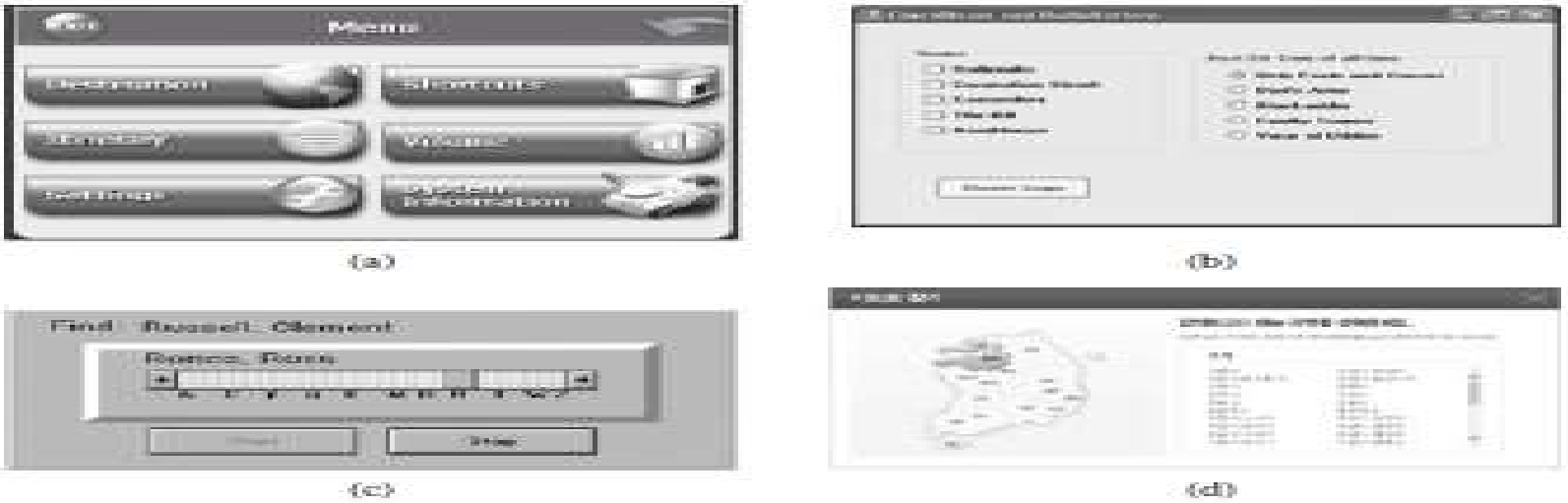
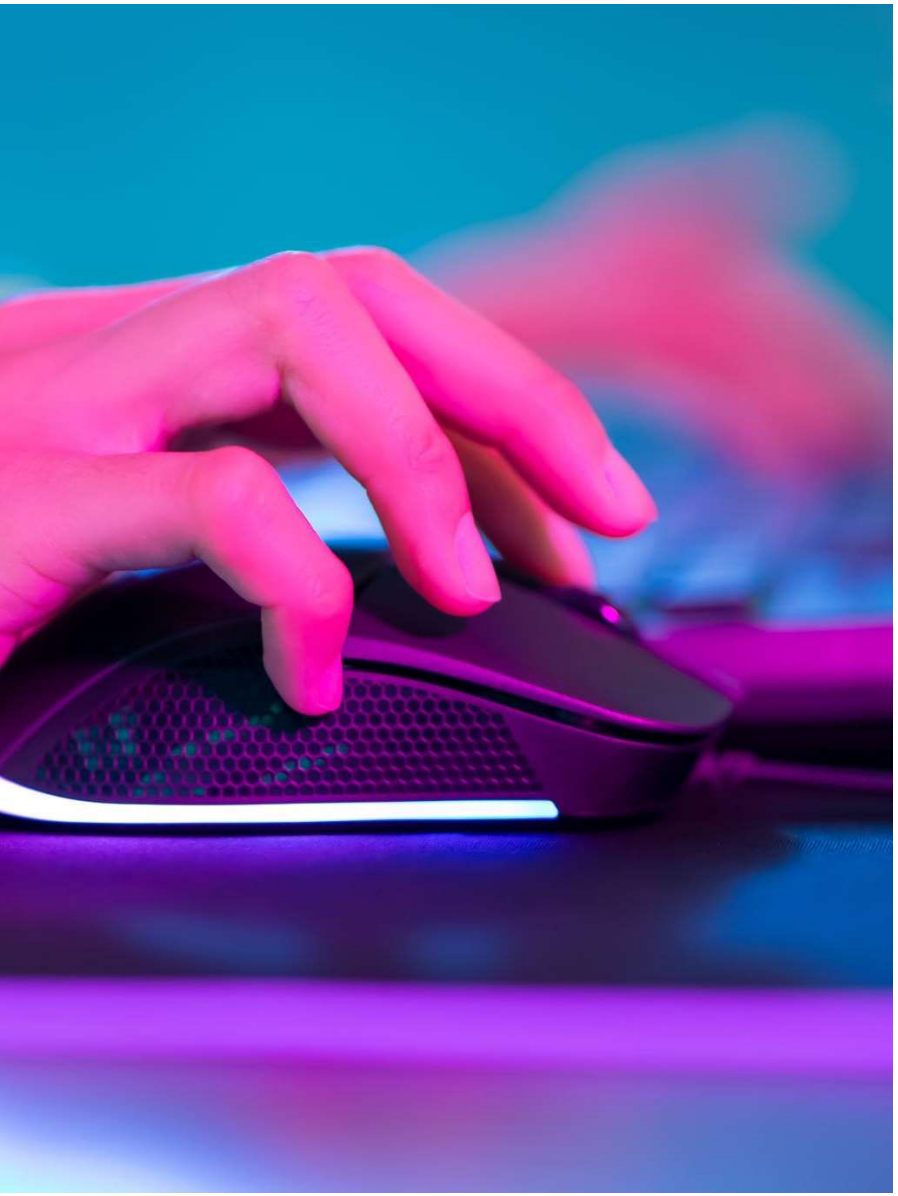


Figure 4.13 Different styles of menus 2: (a) buttons, (b) check boxes and radio buttons, (c) slider menu, (d) image map.

Table 4.1 Where to Use Different Menu Styles

MENU TYPE	USAGE
Pull down	Top level (main) categorical menu
Pop up	Object specific, context specific
Toolbar	Functional/operational tasks
Tabs	File folder metaphor (categorical menu)
Scroll menu	Long menu (many menu items)
2-D array/Image maps	Identification of items by icons (vs. by long names) or pictures
Buttons/Hyperlinks	Short menu (few choices)
Check boxes/Radio buttons	Multiple choice/exclusive choice
Hot keys	For expert users
Aural menu	Telemarketing and for use by the disabled



Software Interface Components

- **Direct interaction:** The mouse/touch-based interaction is strongly tied to the concept of direct and visual interaction.
- Before the mouse era, the HCI was mostly in the form of keyboard inputting of text commands.
- The mouse made it possible for users to apply a direct metaphoric “touch” upon the target objects (which are visually and metaphorically represented as concrete objects with the icons) rather than “commanding” the operating system (via keyboard input) to indirectly invoke the job.

Software Interface Components

- **GUI components**: Software interaction objects are mostly visual.
- The windows, icons, menus, and mouse/pointer-based interactions, which are the essential elements for the graphical user interface (GUI), also sometimes referred to as the WIMP (window, icon, mouse, and pointer).
- Text box: Used for making short/medium alphanumeric input
- Toolbar: A small group of frequently used icons/functions organized horizontally or vertically for a quick direct access
- Forms: Mixture of menus, buttons, and text boxes for long thematic input
- Dialog/combo boxes: Mixture of menus, buttons, and text boxes for short mixed-mode input

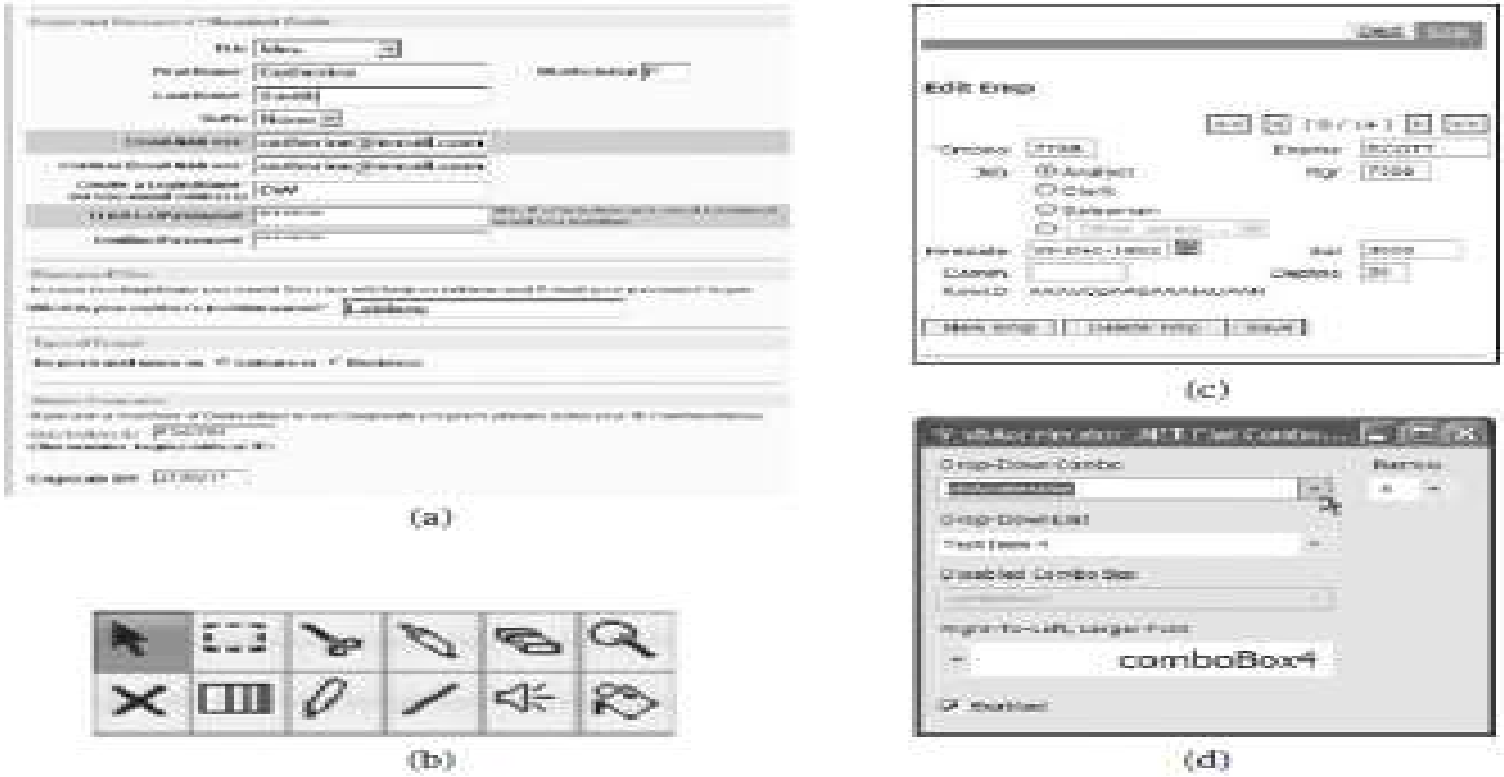
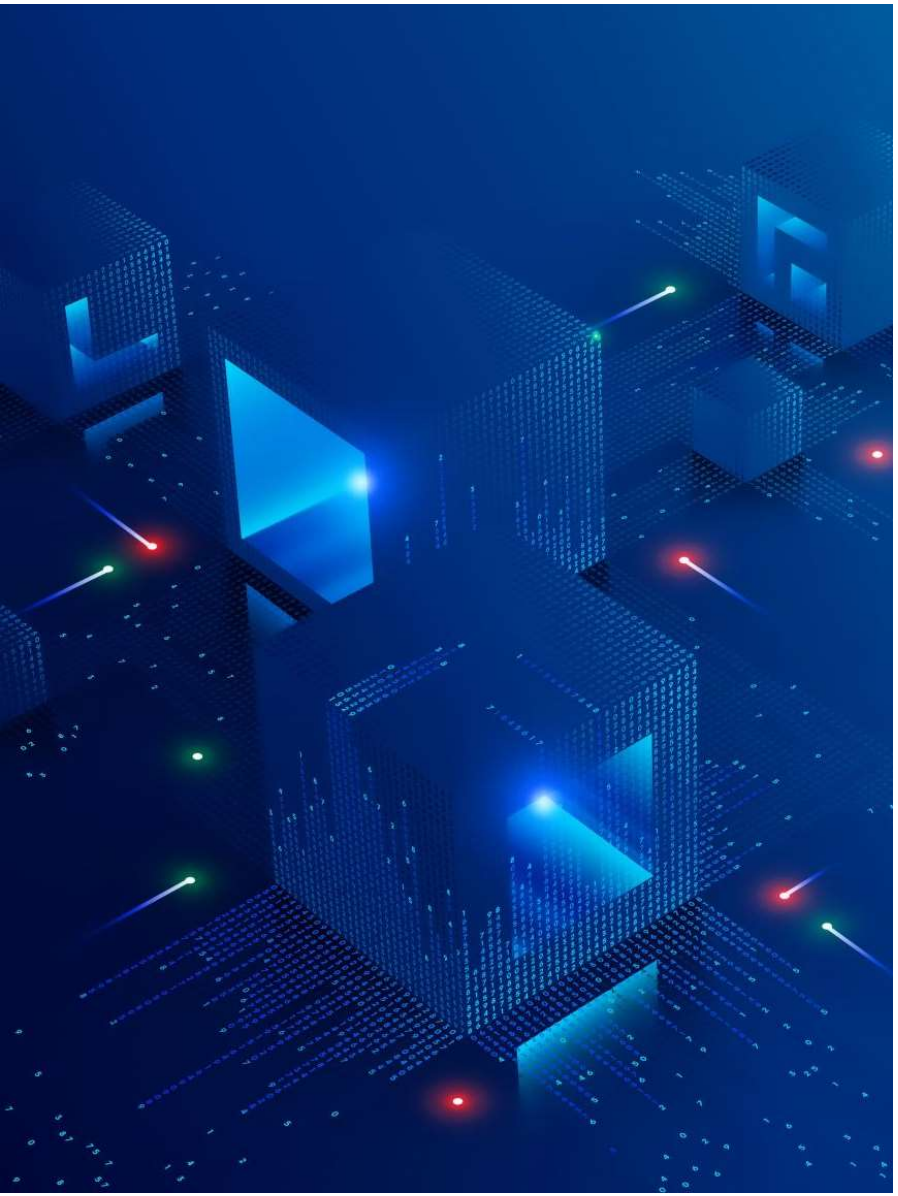
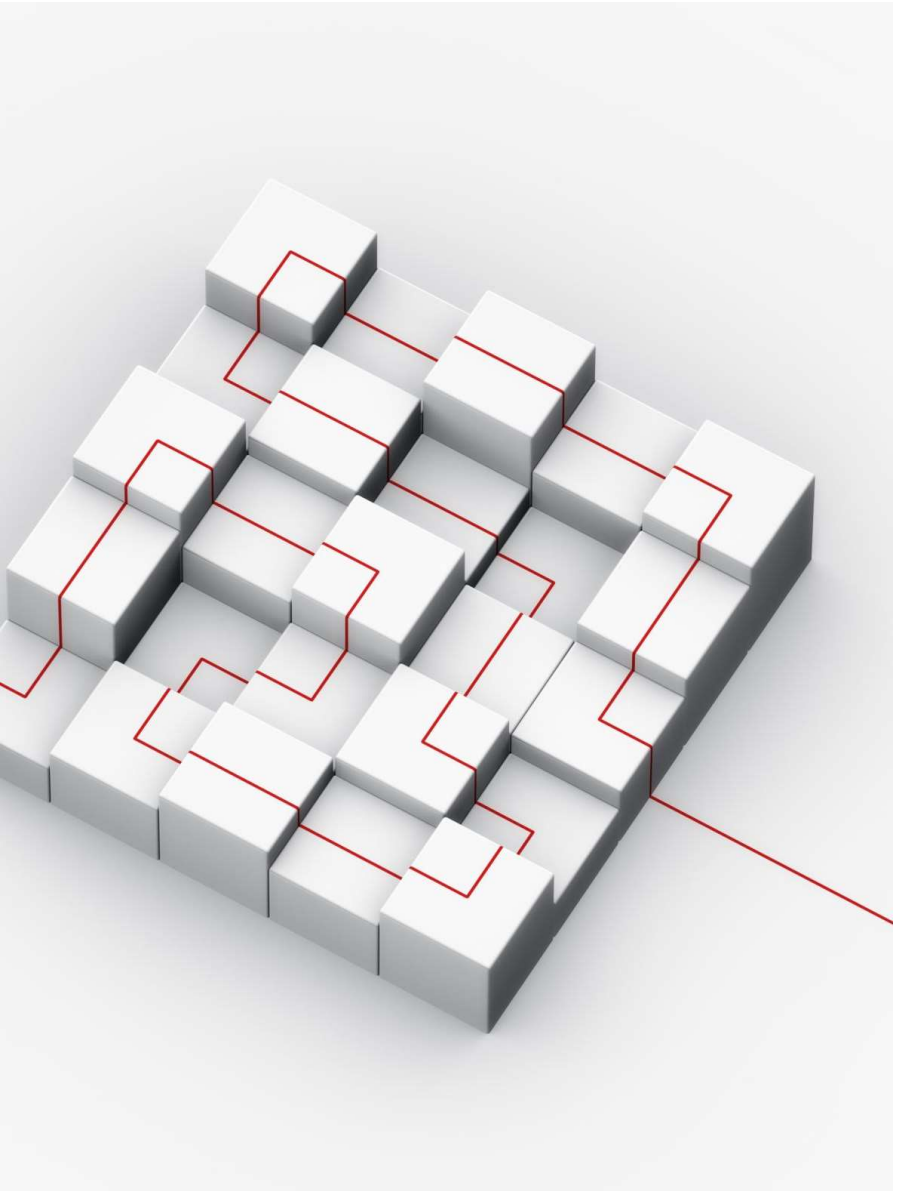


Figure 4.14 GUI interface components: (a) form, (b) toolbar, (c) dialog box, (d) combo box.



Software Interface Components

- 3-D interface (in 2-D interaction input space): Standard GUI elements that are operated and presented in the 2-D space, i.e., they are controlled by a mouse or touch screen and laid out on a 2-D screen



Wire-Framing

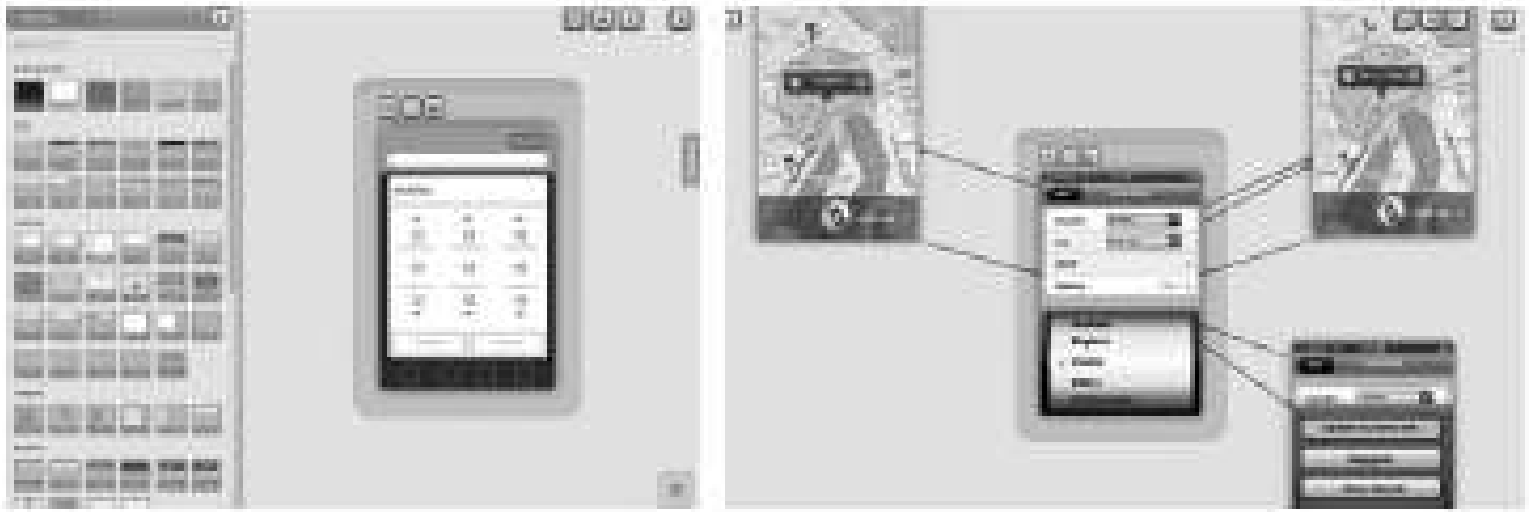
- The interaction modeling and interface options can be put together concretely using the so-called wire-framing process.

- Wire-framing originated from making rough specifications for website page design and resembles scenarios or storyboards.

- Usually, wire-frames look like page schematics or screen blueprints, which serve as a visual guide that represents the skeletal framework of a website or interface

Wire framing

- It depicts the page layout or arrangement of the UI objects and how they respond to each other.
- Wireframes can be pencil drawings or sketches on a whiteboard, or they can be produced by means of a broad array of free or commercial software applications.
- Wireframes produced by these tools can be simulated to show interface behavior, and depending on the tools, the interface logic can be exported for actual code implementation.
- Through wire-framing, the developer can specify and flesh out the kinds of information displayed, the range of functions available, and their priorities, alternatives, and interaction flow.



“Naïve” Design Example:

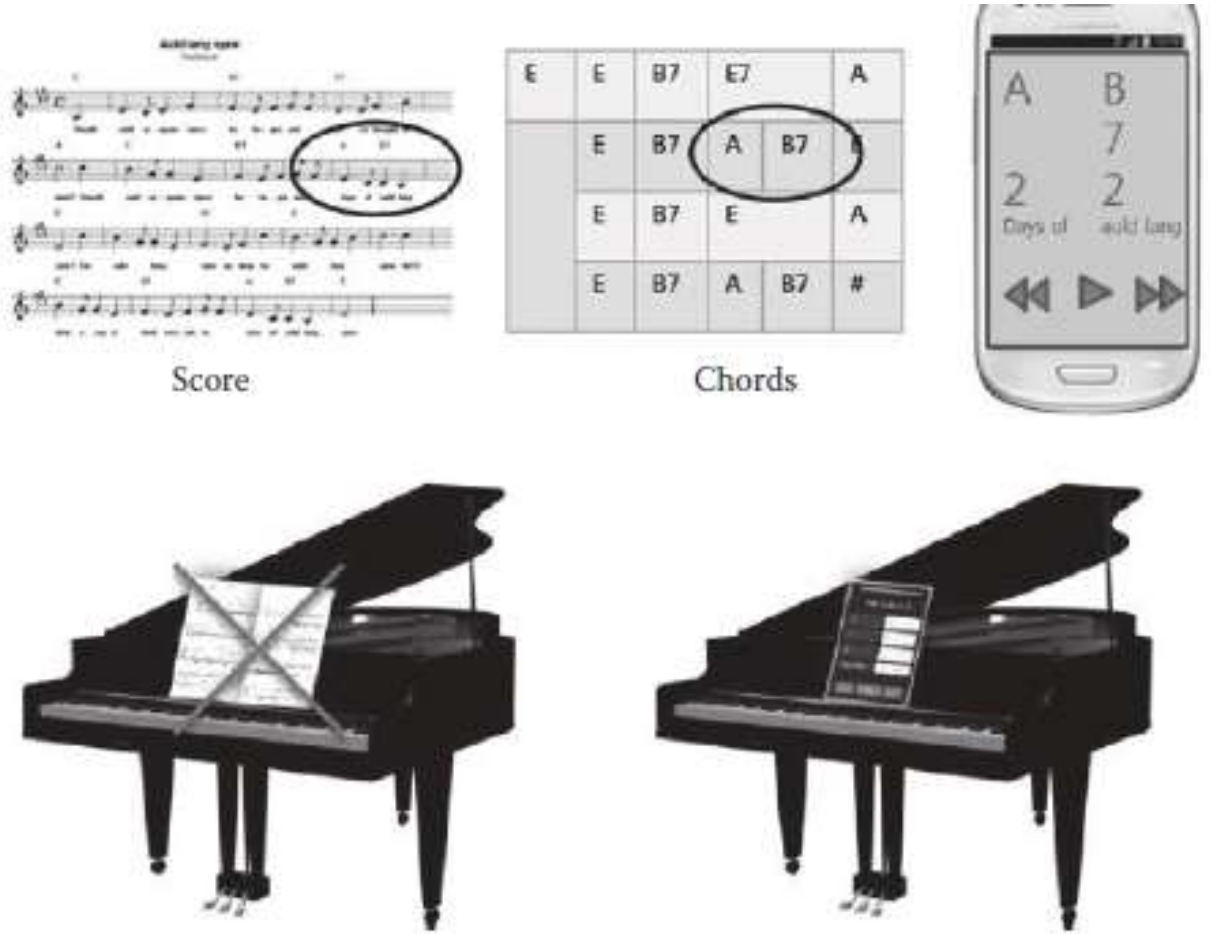


Figure 4.17 No Sheets: Replacing paper sheet music with the smartphone. No more flying pages; no more awkward flipping and page searching.

“Naïve” Design Example:

- ***Requirements Analysis:*** The main purpose of this application is to use the smartphone to present sheet music,* thereby eliminating the need to handle paper sheet music.

1. Use the smartphone to present transcribed music like “sheet music.” Transcription includes only those for basic accompaniment like the chord information (key and type such as C# dom7), beat information (e.g., second beat in the measure).

2. Eliminate the need to carry and manage physical sheet music. Store music transcription files using a simple file format.

3. Help the user effectively accompany the music by timed and effective presentation of musical information (e.g., paced according to a preset tempo).

4. Help the user effectively practice the accompaniment and sing along through flexible control (e.g., forward, review, home buttons).

5. Help user sing along by showing the lyrics and beats in a timed fashion

Naïve" Design Example

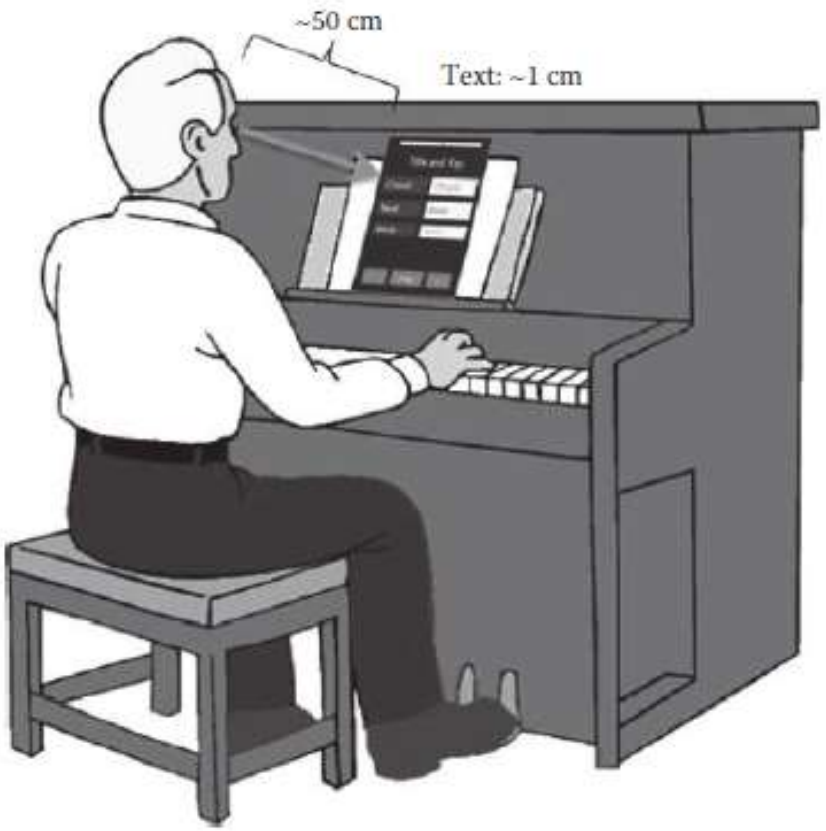
User Analysis

- The typical user for No Sheets is a smartphone owner and novice/ intermediate piano player (perhaps someone who wants to show off their musical skill at a piano bar)
- (e.g., a viewing distance of about 50 cm subtending a letter of —1 cm).

Table 4.3 User Interface Requirements from a Very Minimal User Analysis

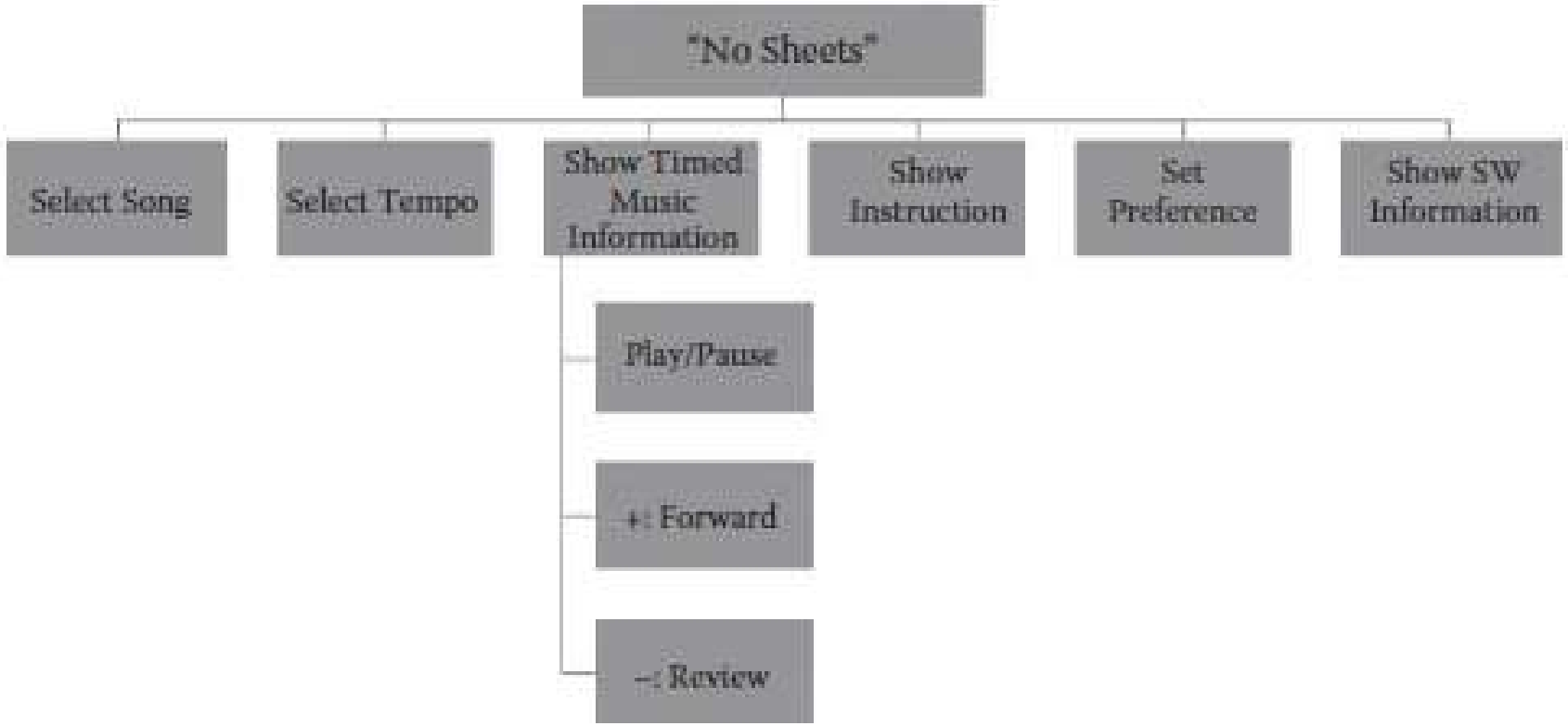
Display mode	Portrait
Layout	Top: Song title Middle: Chord – Beat – Lyrics Bottom: Control buttons
Paging	Left to right Current chord/music info in the left Next chord/music info in the right
Colors	Current chord: Yellow with blue background Next chord: Reversed Buttons: Red Background: Black

User Analysis

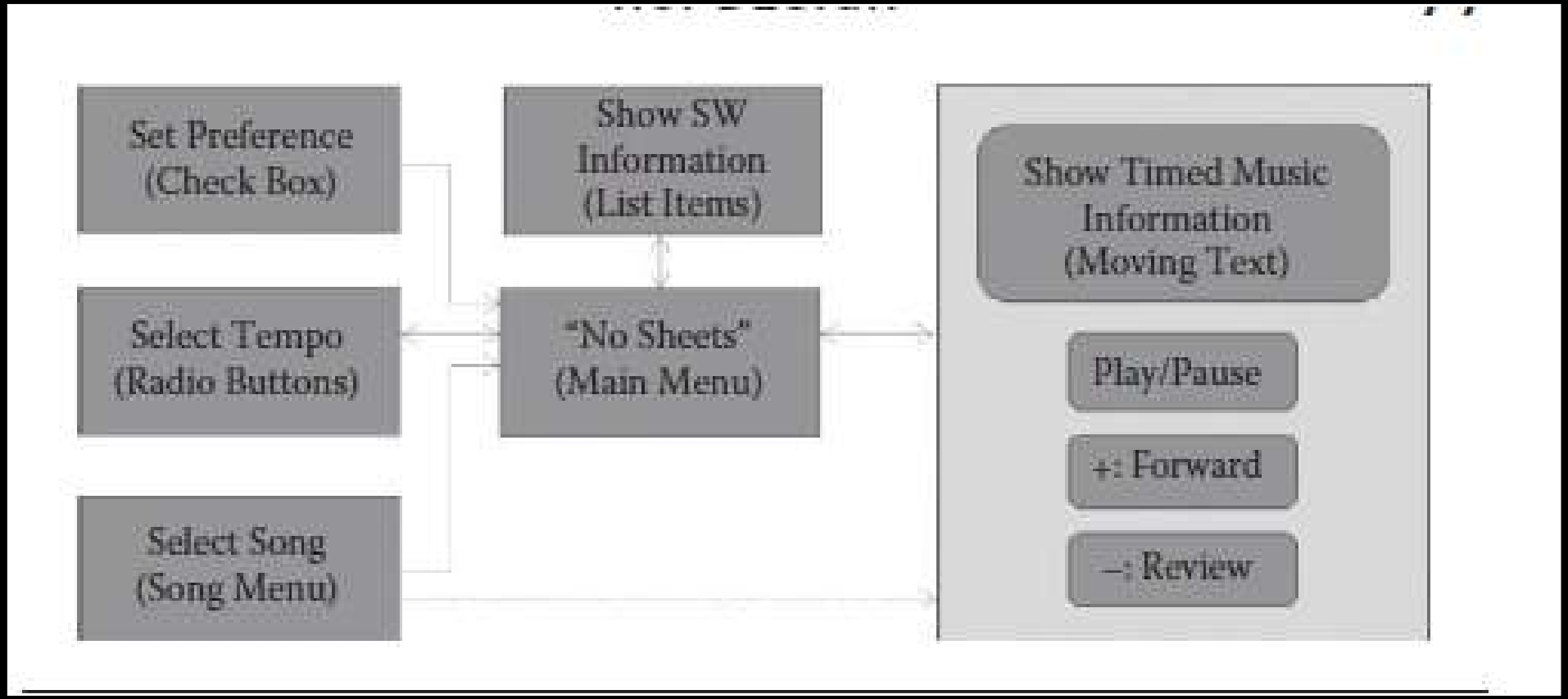


Naïve” Design Example

- ***Making a Scenario and Task Modeling :*** Each task I to be activated directly by the user through an interface.
- ***Select song***: Select the song to view
- ***Select tempo***: Set the tempo of the paging
- ***Show timed music information***: Show the current/next chord/ beat/lyric
- **Play/Pause**: Activate/deactivate the paging
- **Fast-forward:** Manually move forward to a particular point in the song
- **Review:** Manually move backward to a particular point in the song
- ***Show instruction*:** Show the instruction as to how to use the system
- ***Set preferences***: Set preferences for information display and others.
- ***Show software information***: Show version number and developer information


"No Sheets"
Select Song
Select Tempo
Show Timed Music Information
Show Instruction
Set Preference
Show SW Information
Play/Pause
+: Forward
-: Review

Task Model

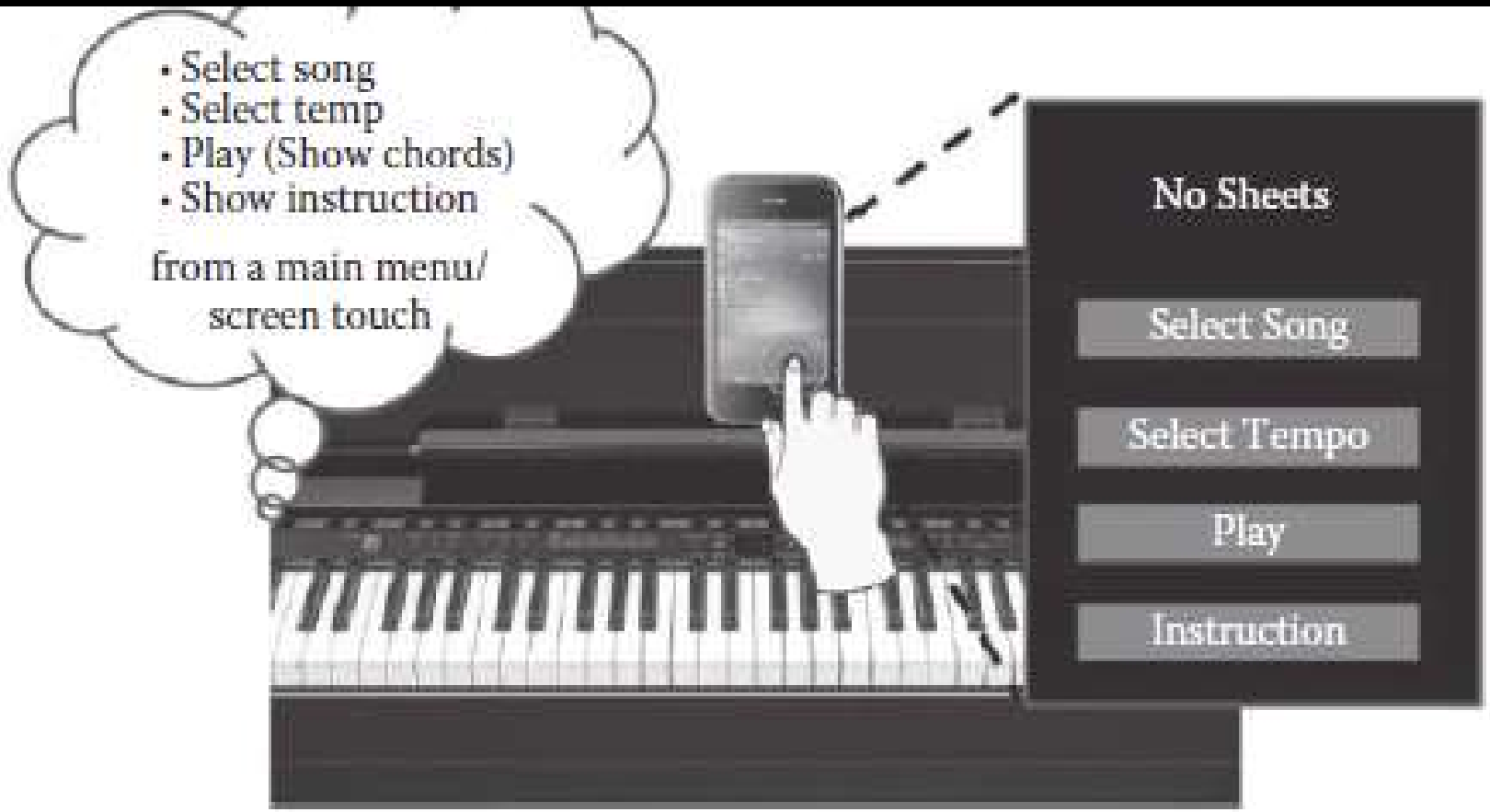


State Transition Diagram

Interface Selection and Consolidation

Table 4.4 Initial Finalization of the Interface Design Choice for No Sheets

SUBTASK	INTERFACE DESIGN CHOICE	JUSTIFICATION
Invoking main functions	• Touch menu • Menu items in red	• Familiar interface • Catch attention
Selecting/changing song	• Scrolling menu • Return to main menu upon selection	• There may be many songs
Selecting/changing tempo	• Scrolling radio buttons • Return to main menu by OK button	• Only one tempo is chosen at a given time
Showing instruction	• Show a one page/screen image with condensed instructional content	• Present condensed content
Playing/pause (view)	• Show progress bar on top • Control interface in the bottom • Provide sound beeps and vibration for first and second beat • Color-code different types of information	• Show status • Familiar interface • Use multimodal feedback for redundancy
Moving forward (+)	• Forward button on the right	• Cultural consideration (moving from left to right) • Show status through progress bar
Moving backward (–)	• Backward button on the left	• Cultural consideration (moving from left to right) • Show status through progress bar
Quitting	• Use platform button	• Use platform (e.g., Android) guideline


• Select song
• Select temp
• Play (Show chords)
• Show instruction
from a main menu/
screen touch
No Sheets
Select Song
Select Tempo
Play
Instruction

- Scroll and select by touch
- Finish by selection, OK, return button

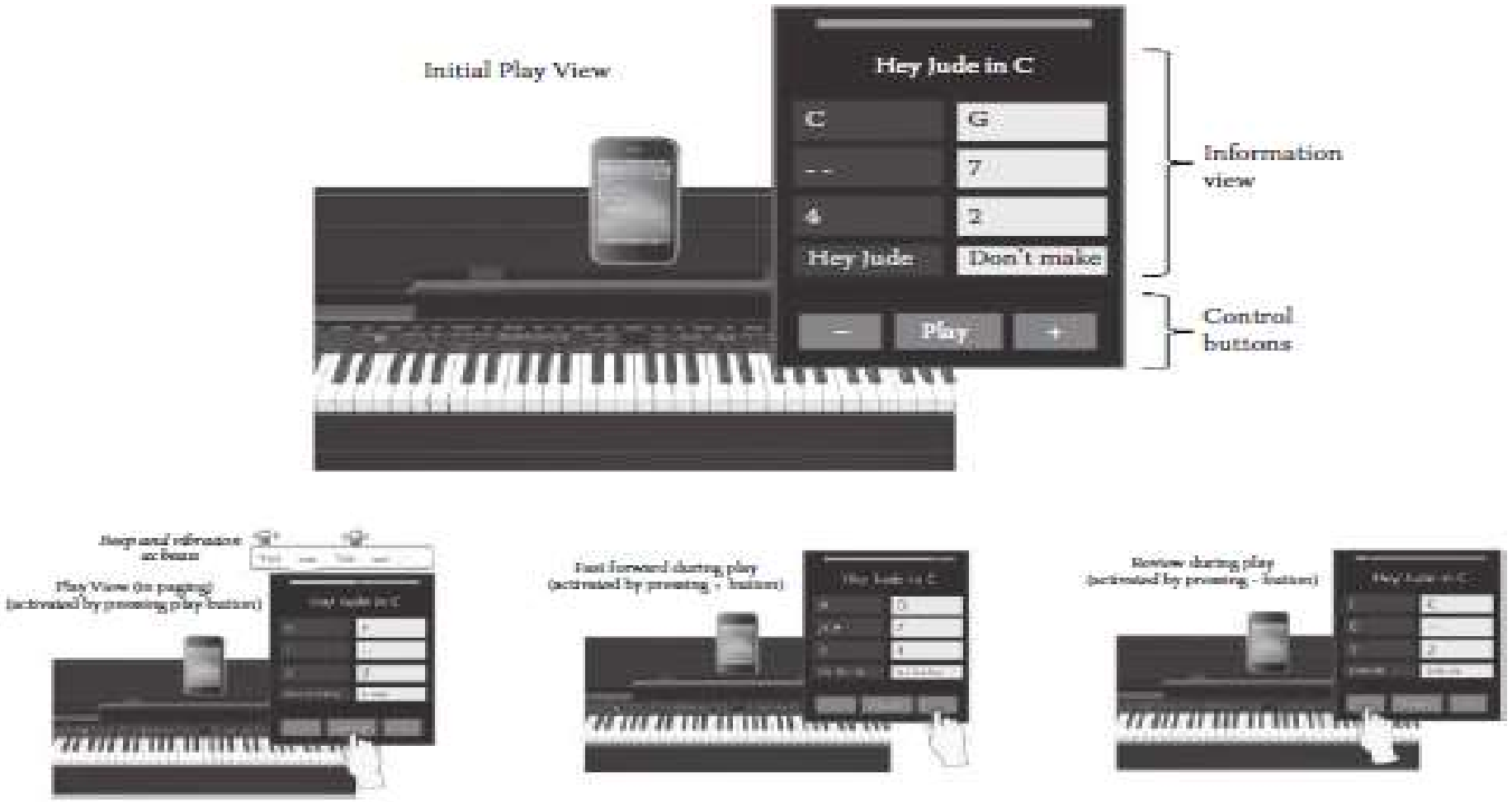


Figure 4.24 A typical usage scene 3: Interface looks during "play" and the three concurrently activatable subtasks (play/pause, move forward, and move backward).

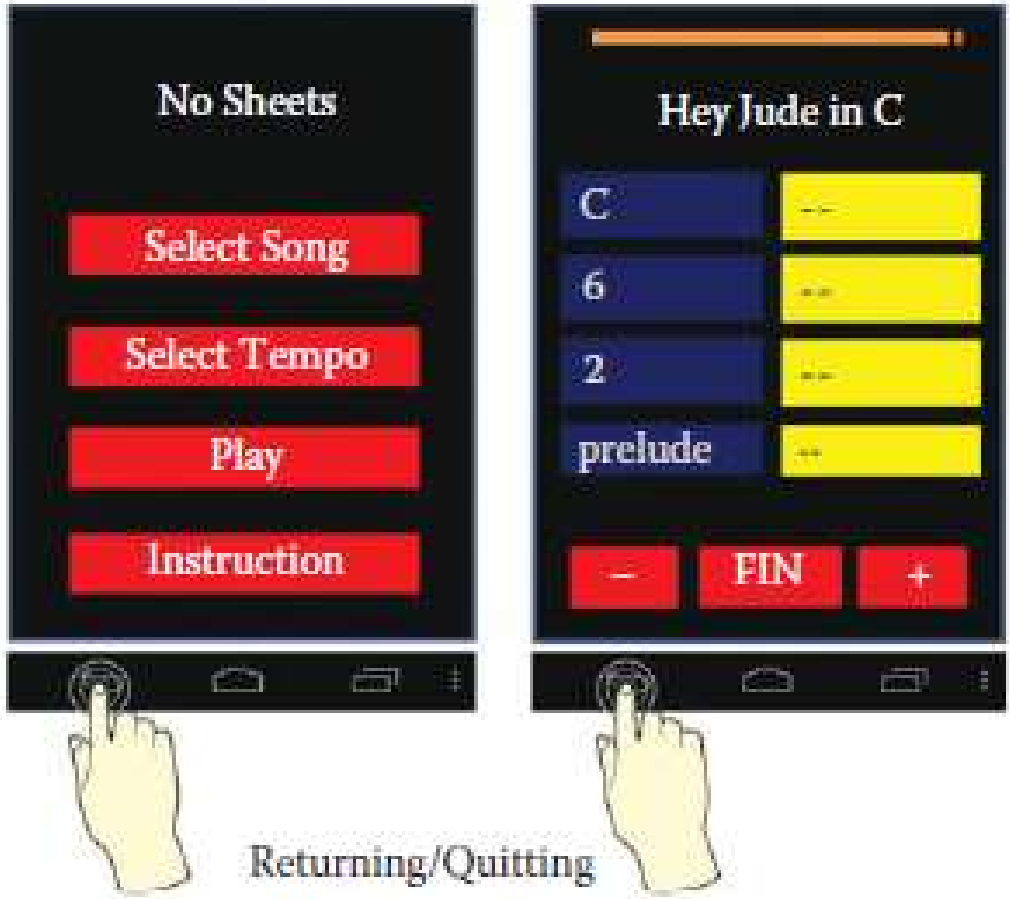


Figure 4.25 A typical usage scene 4: Moving between views/stages and quitting the application by using the standard Android menu button interface.