Human Computer Interaction

Fundamentals and Practice [SWE - 431]

Gerard Jounghyun Kim

Chapter: 4
HCI Design

Mahfuzur Rahman Emon
Lecturer, IICT, SUST

Overall Design Process

HCI design is the process of creating the basic framework for user interaction in software development. It is an 4 steps iterative process.

- **Requirement Analysis:** Any software design starts with a careful analysis of the functional requirements.
    - Functional-Task Requirements: Functions that are to be activated directly by the user through interaction.
      Ex. allow users to create and edit documents in a collaborative manner.
    - Functional-UI Requirements: Functions that are important in realizing certain aspects of the user experience, even though these may not be directly activated by the user.
      Ex. automatic functional feature of adjusting the display resolution of a streamed video based on the network traffic.
    - Nonfunctional-UI Requirements: UI features (rather than computational functions) that are not directly related to accomplishing the main application task.
      Ex. requiring a certain font size or type according to a corporate guideline may not be a critical functional requirement, but a purely HCI requirement feature.

Human Computer Interaction
Chapter 4:

- **User Analysis:** It is simply a process to reinforce the original requirements analysis to further accommodate the potential users in a more complete way. The results of the user analysis will be reflected back to the requirements, and this could identify additional UI requirements (functional or non-functional)

  Ex. particular age group might necessitate certain interaction features such as a large font size and high contrast, or there might be need for a functional UI feature to adjust the scrolling speed.

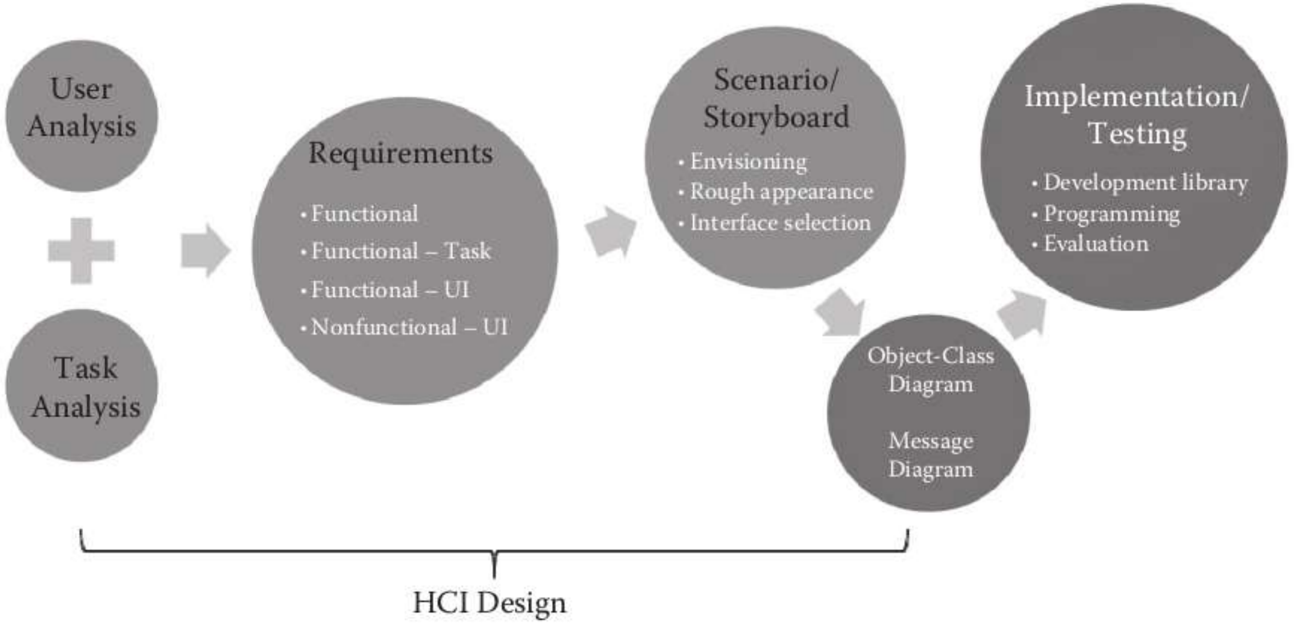
- **Scenario and Task Analysis:** Identifying the application task structure and the sequential relationships between the different elements.

  Through the process of storyboarding, a rough visual profile of the interface can be sketched. Furthermore, the storyboard will serve as another helpful medium in selecting the actual software or hardware interface. It will also serve as a starting point for drawing the object-class diagram, message diagrams, and the use cases for preliminary implementation and programming.

- **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, because not all of these interface components may be available on a working platform.

  Ex. for a particular subtask and application context, the designer might have chosen voice recognition to be the most fitting interaction technique. However, if the required platform does not support a voice sensor or network access to the remote recognition server, an alternative will have to be devised. Such concessions can be made for many reasons besides platform requirements, such as due to constraints in budget, time, personnel, etc.

Human Computer Interaction
Chapter 4:



The overall iterative HCI design process

Interface Selection Options

Hardware Platforms:

- Desktop (Stationary): Monitor, keyboard, mouse, speakers/headphones.
  Suited for: Office related tasks, time consuming/serious tasks, multitasking

- Smartphones/Handhelds (Mobile): LCD Screen, buttons, touch screen, speaker/headphone, microphone, camera, sensors (acceleration, tilt, light, gyro, proximity, compass, barometer), vibrators, mini “qwerty” keyboard
  Suited for: Simple and short tasks, special-purpose tasks

- Tablet/pads (mobile): LCD screen, 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

Human Computer Interaction
Chapter 4:

- Embedded (stationary/mobile): LCD/LED screen, 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.

  Suited for: Special tasks and situations where interaction and computations are needed on the spot (e.g., printer, rice cooker, mp3 player, personal media player.

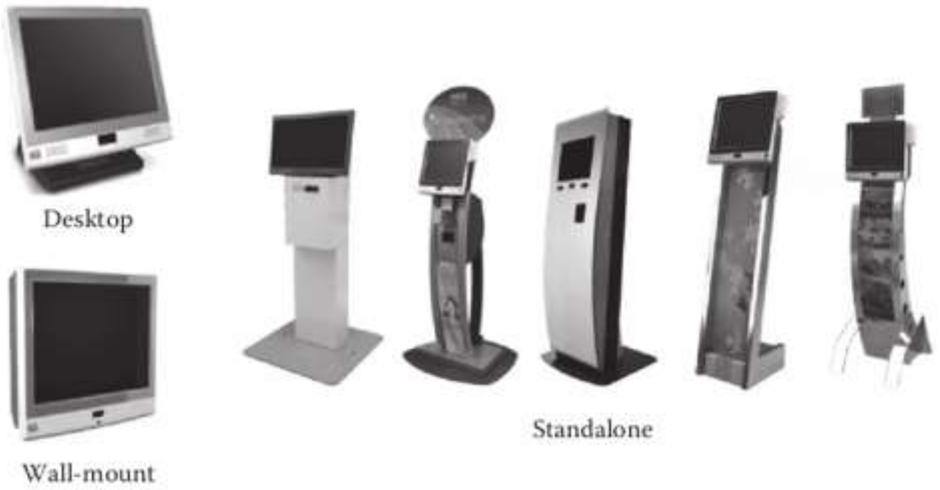
- TV/consoles (stationary): LCD/LED screen, button-based remote control, speaker, microphone, game controller special sensors, peripherals (camera, wireless keyboard, Wii mote–like device [1], depth sensor,

  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, buttons, speaker, touchscreen, 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

Human Computer Interaction
Chapter 4:



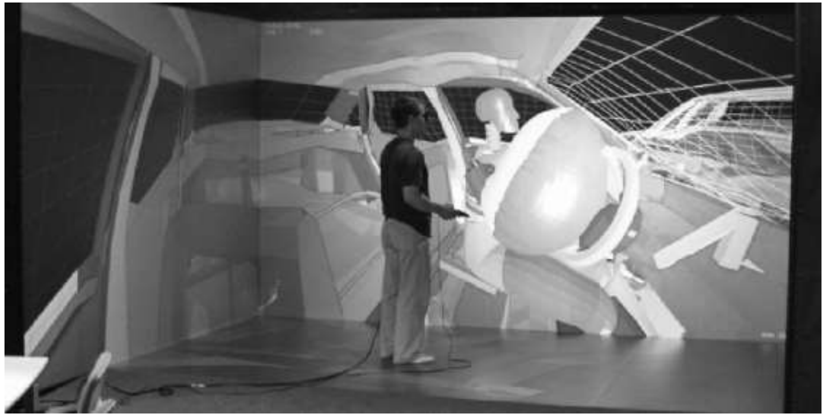
TV/Console interaction environment

Kiosk/installation types of interaction platform.

- Virtual reality (stationary): Large-surround and high-resolution projection screen/head-mounted display/stereoscopic display, 3D tracking sensors, 3-D sound system, haptic/tactile display, special sensors, peripherals (microphone, camera, depth sensors, glove)

  Suited for: Spatial training, tele-experience and tele-presence, immersive entertainment.

- Free form (stationary and mobile): Special-purpose hardware platforms consisting of a customized configuration of individual devices best suited for a given task (when cost is not the biggest factor)



Human Computer Interaction
Chapter 4:

(a)

(b)

(c)

(d)

(e)

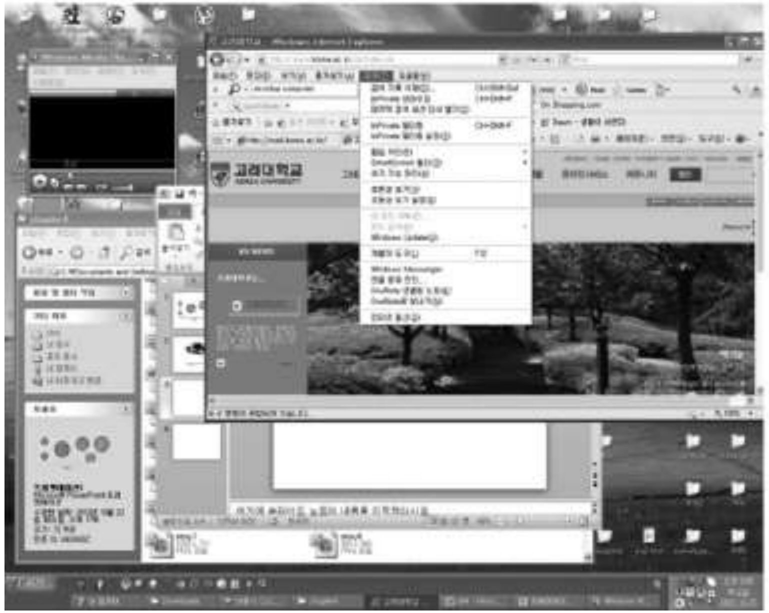
Examples of special-purpose interfaces

(a)pen tablet
(b)a glass-type see-through heads-up display
(c)camera-integrated scuba gear
(d) special military helmet for tactical command and control
(e) multitouch tabletop platform for multiple users

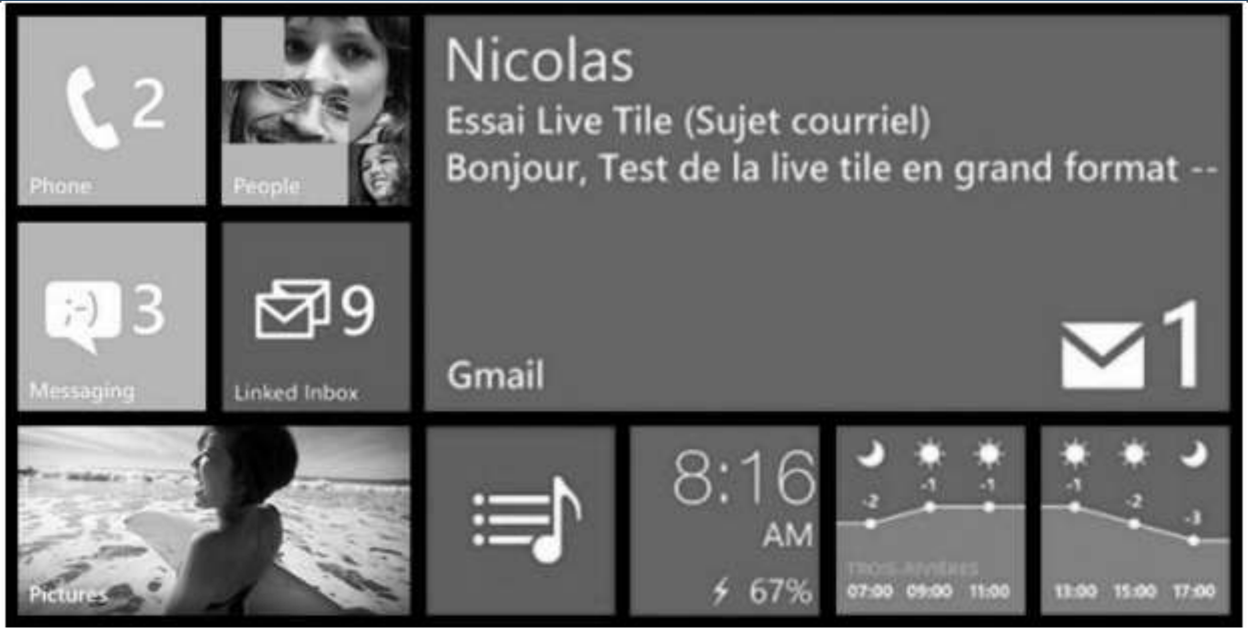
Software Interface Components

Windows/layers:

- Modern desktop interfaces use windows as visual output channels for computational processes.
- Large displays often have overlapping windows, while smaller devices use non overlapping layers activated by gestures.
- The Windows Metro-style interface unifies these approaches, presenting full-screen applications without marked borders.
- Considerations for windows include size, layout, and management methods for interaction.



The hallmark of modern desktop user interfaces: multiple overlapping windows (left) and non overlapping layers for smaller displays (right).



The Microsoft® Metro–style interface that unifies the mobile and desktop interaction.

Icons:

- Interactable objects, like icons, serve as compact visual representations for easy interaction.
- Clickable icons, simple and intuitive, are designed to be informative despite their small size.
- The Windows Metro-style interface introduces dynamic "tiles" that can change appearance, providing useful information such as the number of unread emails within the icon.

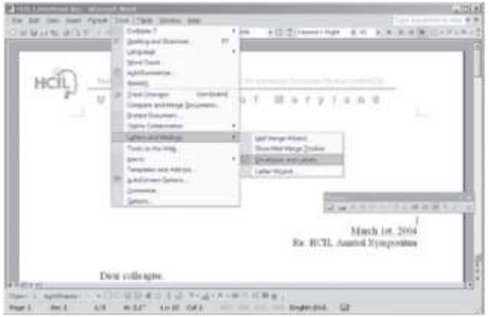
(Left) Obscure Google Chrome browser icon design (difficult to tell at a glance what the icon represents) vs. (right) informative icon design for the Angry Birds application (the visual imagery of an “angry bird” is well captured).

Menu:

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

Different styles of menus

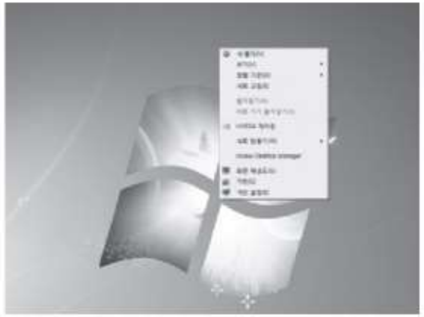
(a) pull down
(b) pop up
(c) 2-D application bar
(d) 1-D toolbar
(e) tabs.

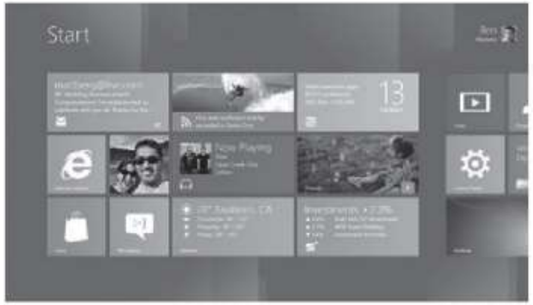

HCIL

March 1st, 2004
Re: HCIL Annual Symposium

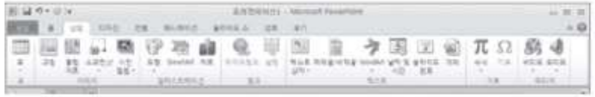
Dear colleagues

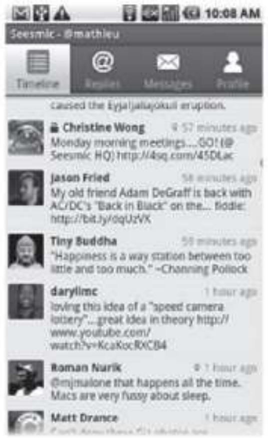
(a)


(b)



(c)

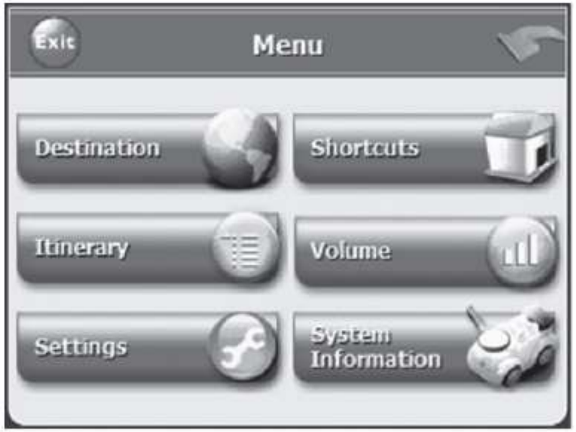

(d)


10:08 AM
Seesmic - @mathieu

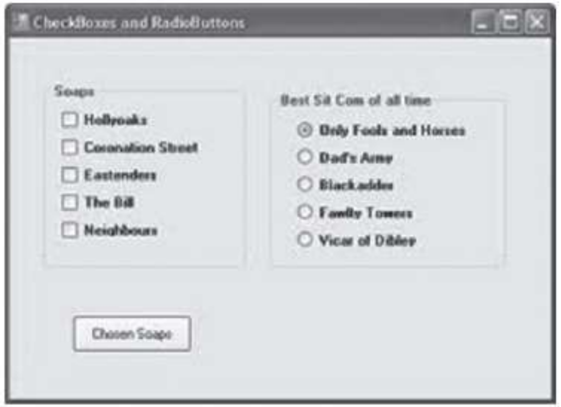
caused the Eyjafjallajokull eruption.
Christine Wong 57 minutes ago
Monday morning meetings....GO! (@ Seesmic HQ) http://4sq.com/45DLac
Jason Fried 58 minutes ago
My old friend Adam DeGraff is back with AC/DC's "Back in Black" on the... fiddle: http://bit.ly/dqUzVX
Tiny Buddha 58 minutes ago
"Happiness is a way station between too little and too much." ~Channing Pollock
darylimc 1 hour ago
loving this idea of a "speed camera lottery"... great idea in theory http://www.youtube.com/watch?v=KcaKocRXCB4
Roman Nurik 1 hour ago
@mjmalone that happens all the time. Macs are very fussy about sleep.
Matt Drance 1 hour ago

(e)

Human Computer Interaction
Chapter 4:



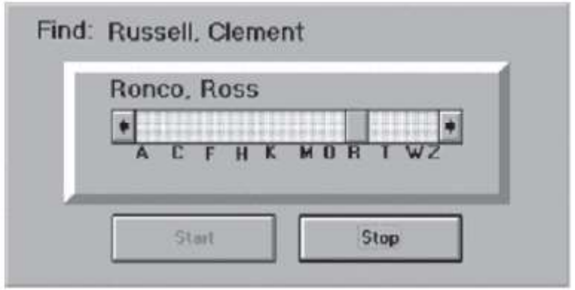
(a)



Soaps
- [ ] Hollyoaks
- [ ] Coronation Street
- [ ] Eastenders
- [ ] The Bill
- [ ] Neighbours

Best Sit Com of all time
- [✓ Only Fools and Horses]
- Dad's Army
- Blackadder
- Fawlty Towers
- Vicar of Dibley

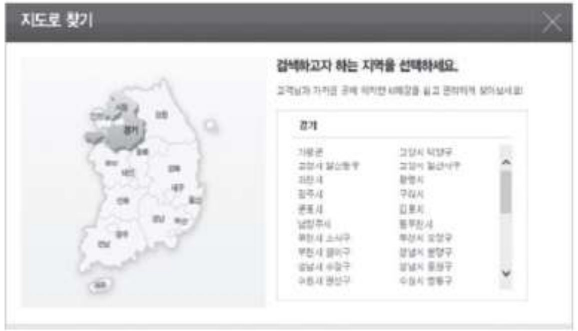
(b)



Find: Russell, Clement

Ronco, Ross

(c)



(d)

Different styles of menus
(a) buttons
(b) checkboxes and radio buttons
(c) slider menu
(d) image map.

Where to use different menu items:

- 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

Menus consist of subtasks or target objects for tasks. They should be organized hierarchically, ideally with fewer than **eight items** per level to accommodate short-term memory limits. If long menus are necessary, systematic layouts based on frequency, importance, or alphabetical order are recommended.

Direct interaction:

The mouse/touch-based interaction revolutionized Human-Computer Interaction (HCI), shifting from text commands with keyboards to direct and visual interaction. Users could now visually interact with icons, enabling direct manipulation like "dragging and dropping," "cutting and pasting," and "rubber banding," enhancing user experience.

GUI Components:

- Often referred to as WIMP (window, icon, mouse, and pointer) interfaces, are crucial for visual software interaction.
- Despite the negative connotation to emphasize its contrast with emerging user interfaces like voice/language and gesture-based systems. WIMP interfaces have played a significant role in the widespread adoption of computer technologies.
- Key GUI components include text box, toolbar, forms, dialog/combo boxes.

3D interface (in 2D interaction input space):

Standard GUI elements operate in 2-D space, controlled by a mouse or touch screen. In 3D applications like games, 2-D control may be insufficient, causing fatigue. Non-WIMP interfaces like 3D motion gestures are gaining popularity for tasks like 3D games. While you can put these 2-D elements into a 3D space, it may not always make things better, especially if you can only see it from one angle on the screen. But in games, just for looks and the "wow" factor, people still sometimes use 3D setups.

Other (non-WIMP) interfaces:

The WIMP interface revolutionized computing in the early 1980s. Advances in technologies like voice and gesture recognition, along with changes in computing environments, are introducing new interfaces into our daily lives. Cloud computing enables running complex interface algorithms on less powerful devices, explored further in Chapters 7–9 of this book.

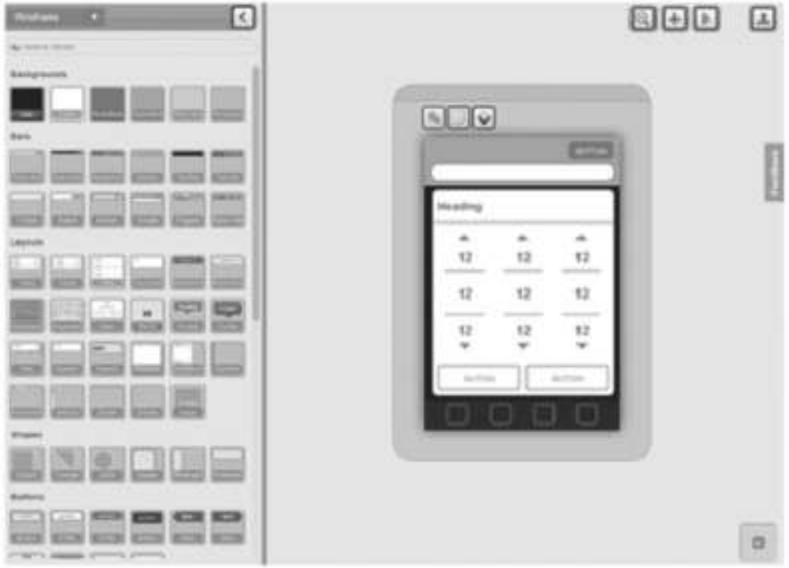
Human Computer Interaction
Chapter 4:

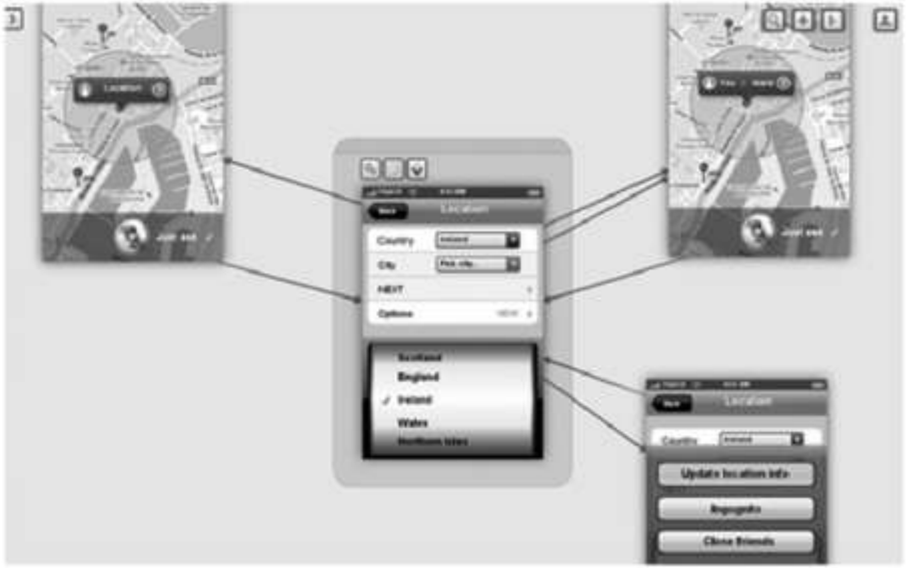
(a)

(b)

Human Computer Interaction
Chapter 4:

Wire Framing: Wire-framing is like sketching a plan for how a website or interface should look and work. It helps developers figure out what information is shown, what functions are available, and how everything interacts. These sketches can be made on paper or with software, and some tools even let you see how the interface would behave. It's important to keep design and implementation separate during this process.

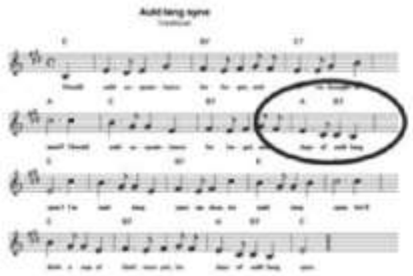




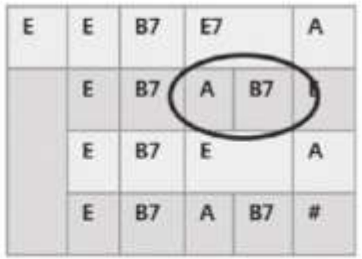
“Naïve” Design Example: No Sheets 1.0

Requirement Engineering: Initial requirements for No Sheets:

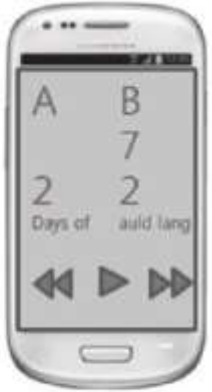
- Use the smartphone to present transcribed music like “sheet music.” Transcription includes only those for basic accompaniment like the chord information (key and type such as C# dom7), beat information (e.g., second beat in the measure).
- Eliminate the need to carry and manage physical sheet music. Store music transcription files using a simple file format.
- Help the user effectively accompany the music by timed and effective presentation of musical information (e.g., paced according to a preset tempo).
- Help the user effectively practice the accompaniment and sing along through flexible control (e.g., forward, review, home buttons).
- Help user sing along by showing the lyrics and beats in a timed fashion



Score



Chords

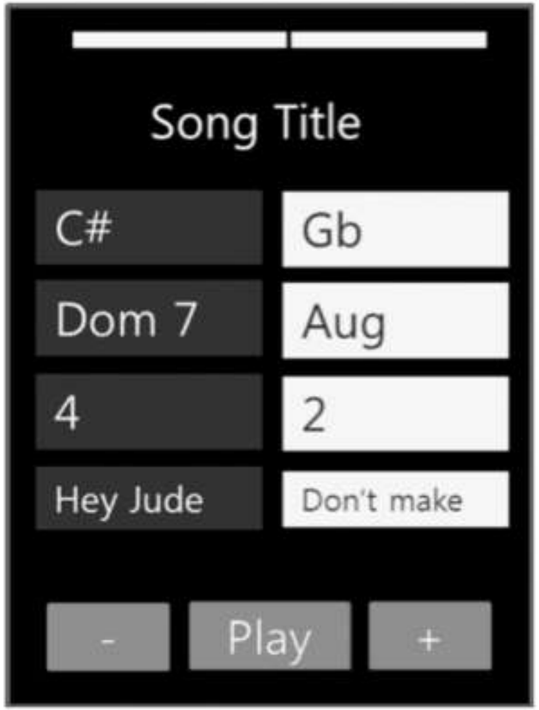


No Sheets: Replacing paper sheet music with the smartphone. No more flying pages; no more awkward flipping and page searching.

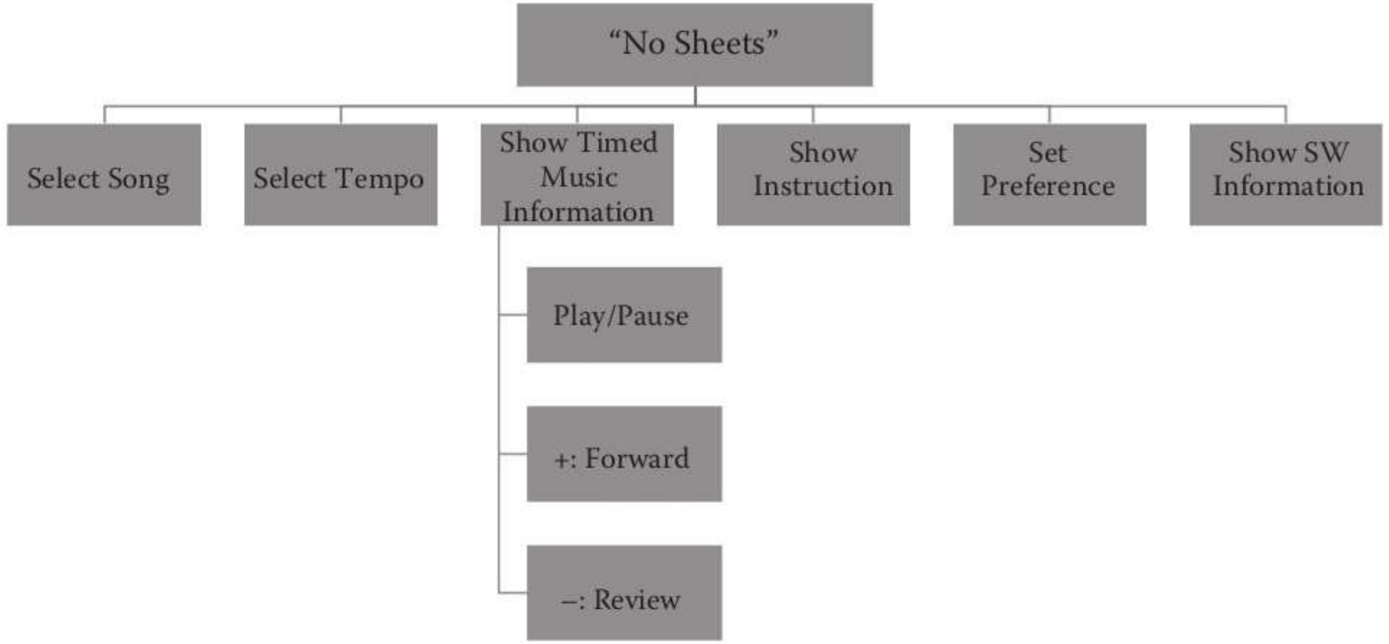
User Analysis:

- The typical No Sheets user is a smartphone owner and a novice to intermediate piano player, potentially looking to showcase musical skills at a piano bar.
- The expectation is for reasonable eyesight, considering a smartphone viewing distance of around 50 cm.
- No specific age or gender group is targeted.
- Interface preferences may exist regarding chord/music display, such as portrait vs. landscape orientation, layout, control button placement, color-coding, and scrolling methods.
- The initial interface requirements, based on the developer's trial, are listed in next page, a more thorough user analysis is planned for a revised design in Chapter 8.

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



Scenario and Task Analysis



A simple task model for No Sheets. The top-level application has six subtasks (select song, select tempo, etc.), and the third subtask (show music info) has yet other subtasks: play/pause, fast-forward, and review.

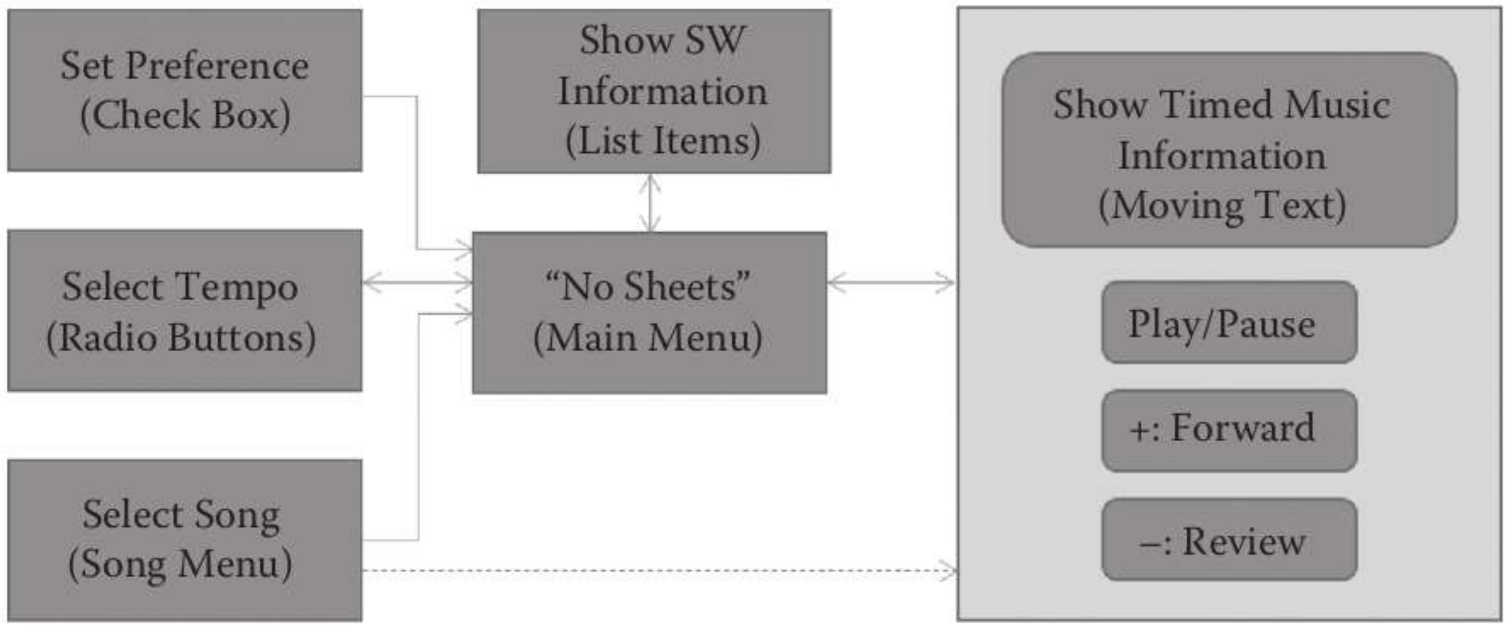
- Select tempo: Set the tempo of the paging

- 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

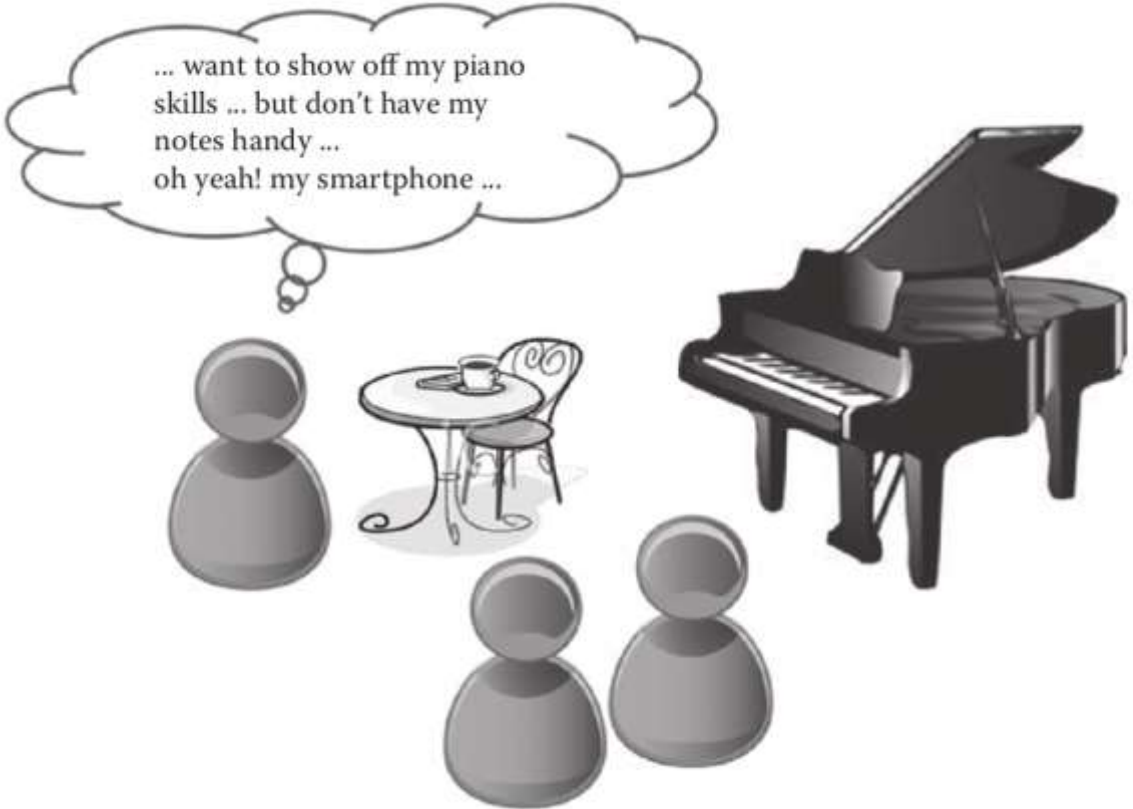
- Set preferences: Set preferences for information display and others

- Show software information: Show version number and developer information



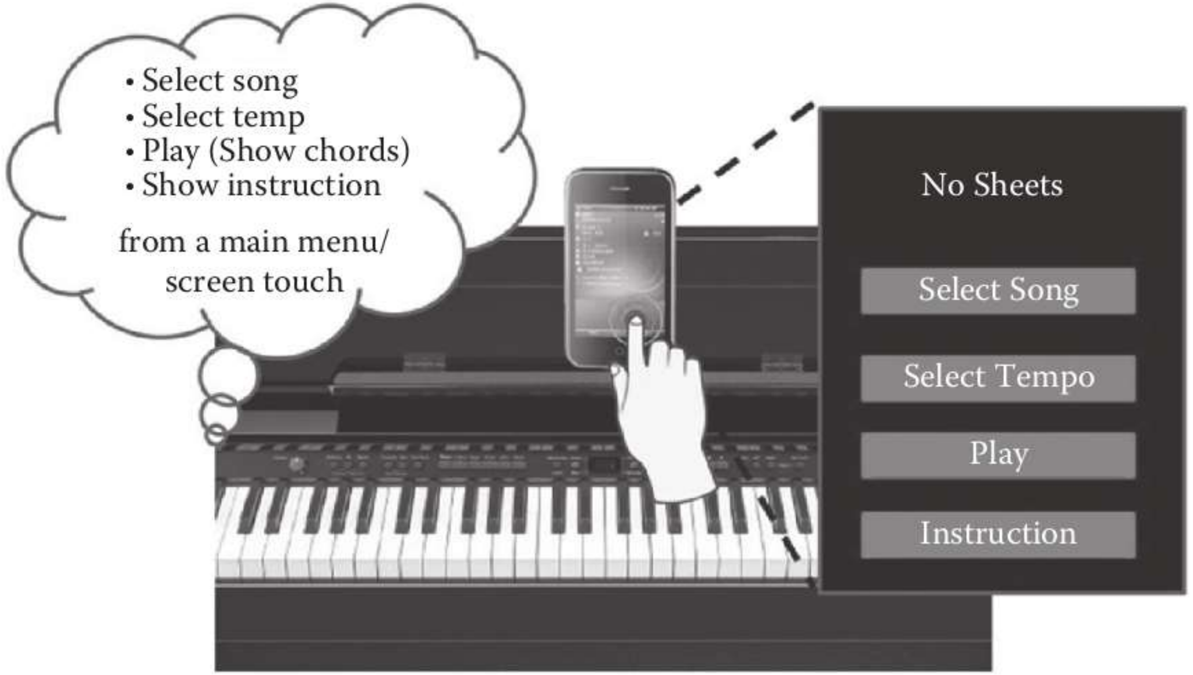
A possible state transition diagram

Storyboards from task model


... want to show off my piano skills ... but don't have my notes handy ... oh yeah! my smartphone ...

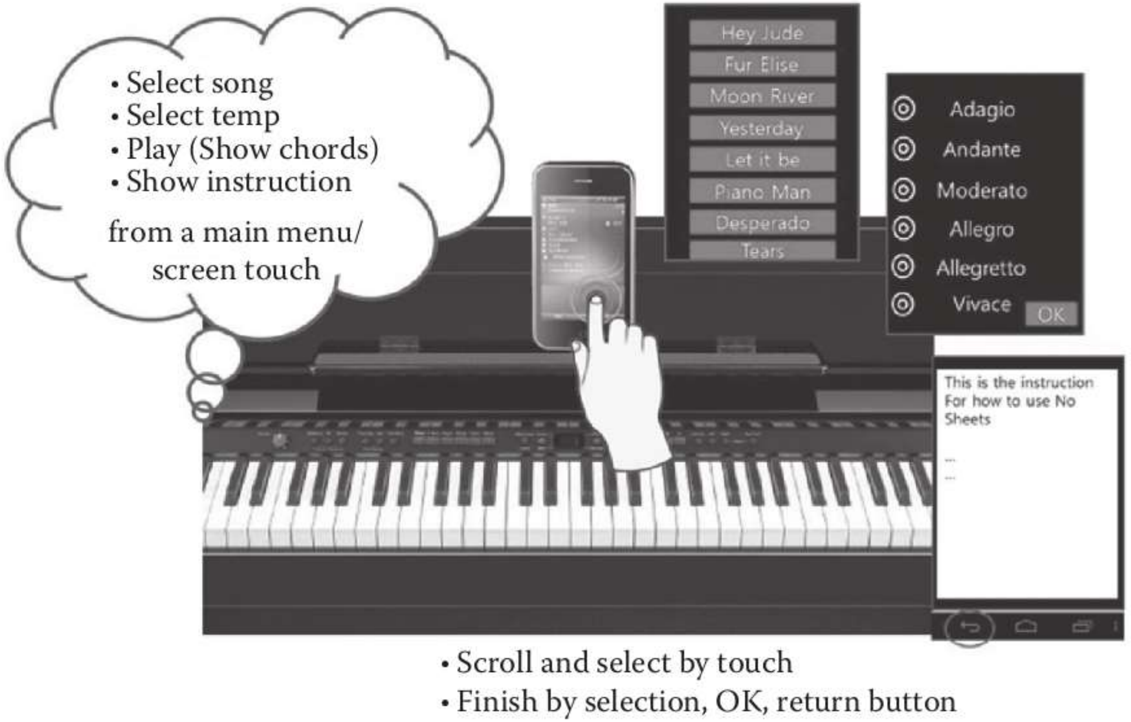
Motivational context for No Sheets.

Storyboards from task model



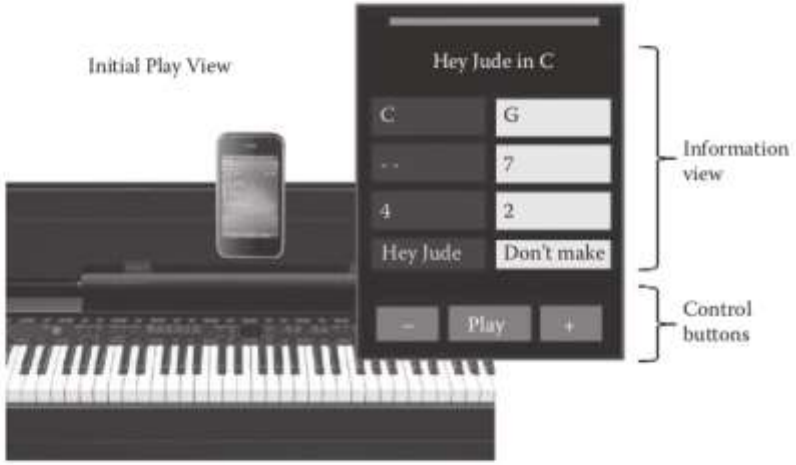
A typical usage scene 1: The top-level menu

Storyboards from task model



A typical usage scene 2: Interface looks for three subtasks (e.g., song selection, tempo selection, and showing instruction).

Storyboards from task model



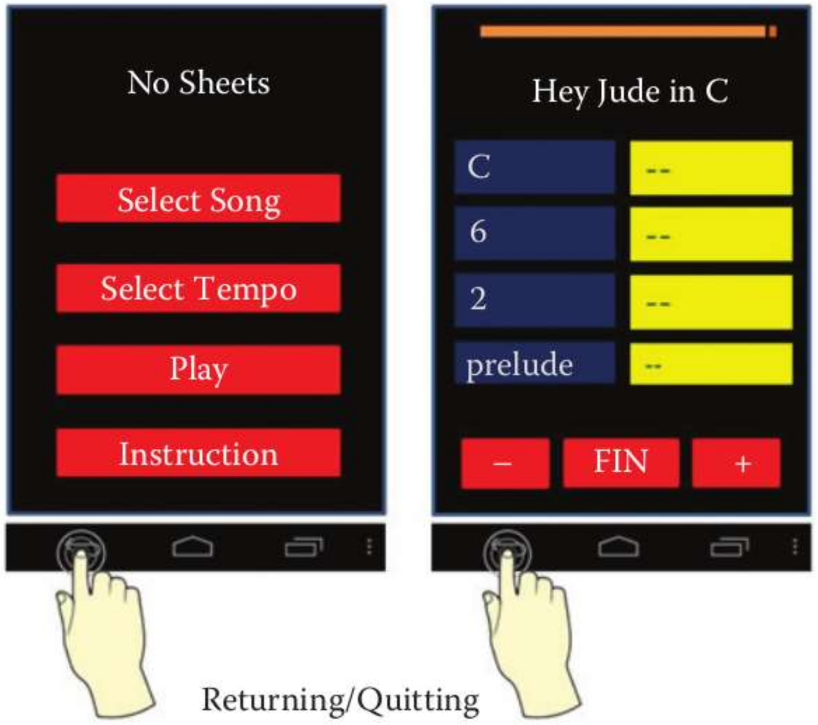
A typical usage scene 3: Interface looks during "play" and the three concurrently activatable subtasks (play/pause, move forward, and move backward).

Beep and vibration at beats
Play View (in paging)
(activated by pressing play button)
Hey Jude in C

Fast forward during play
(activated by pressing + button)
Hey Jude in C

Review during play
(activated by pressing - button)
Hey Jude in C

Storyboards from task model

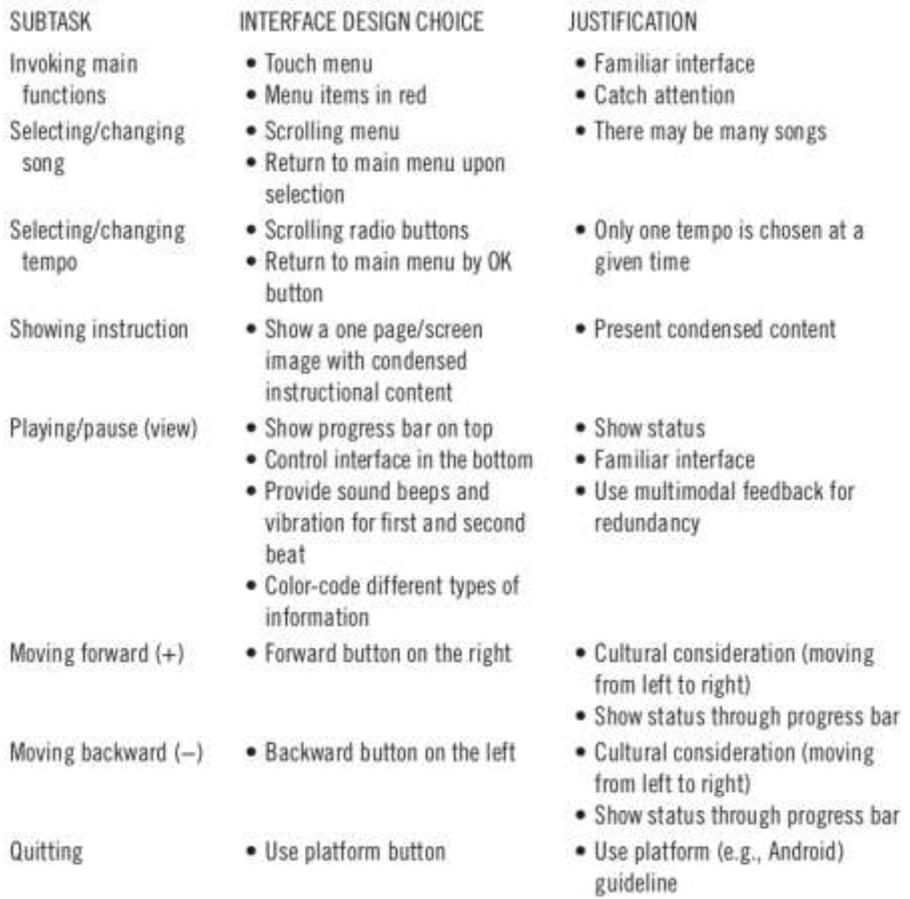


A typical usage scene 4: Moving between views/stages and quitting the application by using the standard Android menu button interface.

Interface Selection and Consolidation

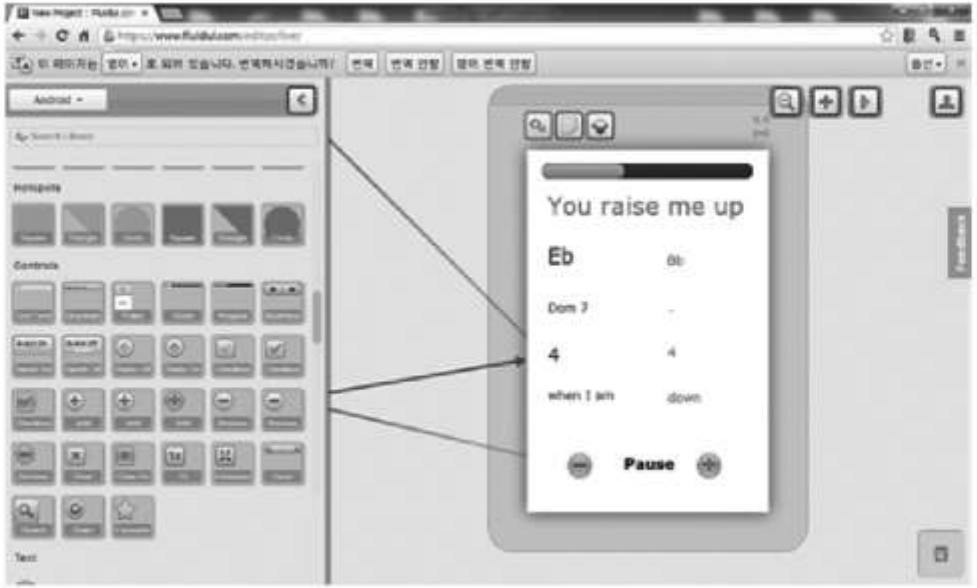
- We try to adhere to the HCI principles, guidelines, and theories to justify and prioritize our decision.
- Our initial choice will purposely not be well thought out, just to illustrate that a naïve and hurried choice would expose the application to be at high risk of eventually failing in terms of usability and user experience, even if it computationally satisfies the required functionalities.
- This will become more apparent as we evaluate the initial prototype and revise our requirements and design for No Sheets 2.0

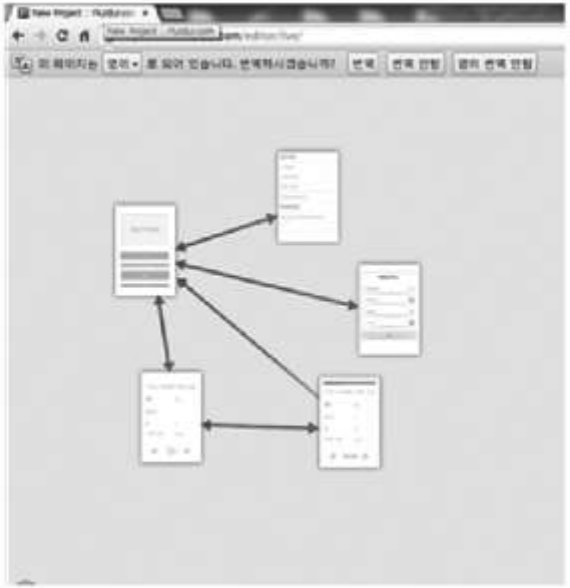
Human Computer Interaction
Chapter 4:



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

Human Computer Interaction
Chapter 4:





Initial design wireframe for No Sheets 1.0 using a wire-framing tool.
Left: Icons and GUI elements in the menu in the left can be dragged onto the right to design the interface layer.
Right: Navigation among the design layers can be defined as well (indicated by the arrows).