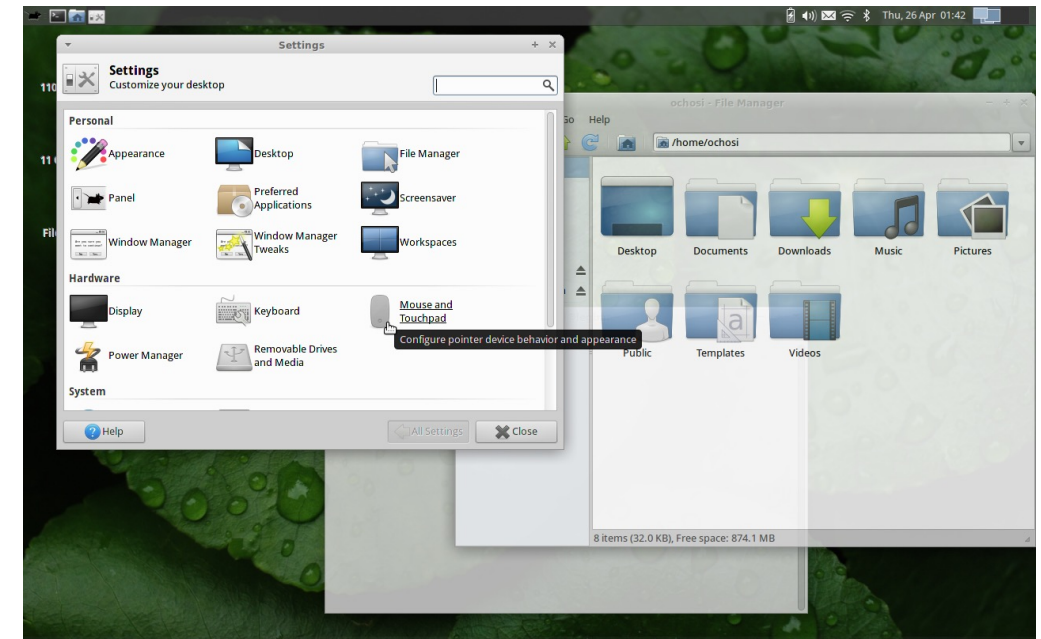


# The User Interface (UI)



By Xfce Development Team - <http://cdn.xfce.org/about/screenshots/4.10-1.png> (Screenshot gallery), GPL, <https://commons.wikimedia.org/w/index.php?curid=19296500>

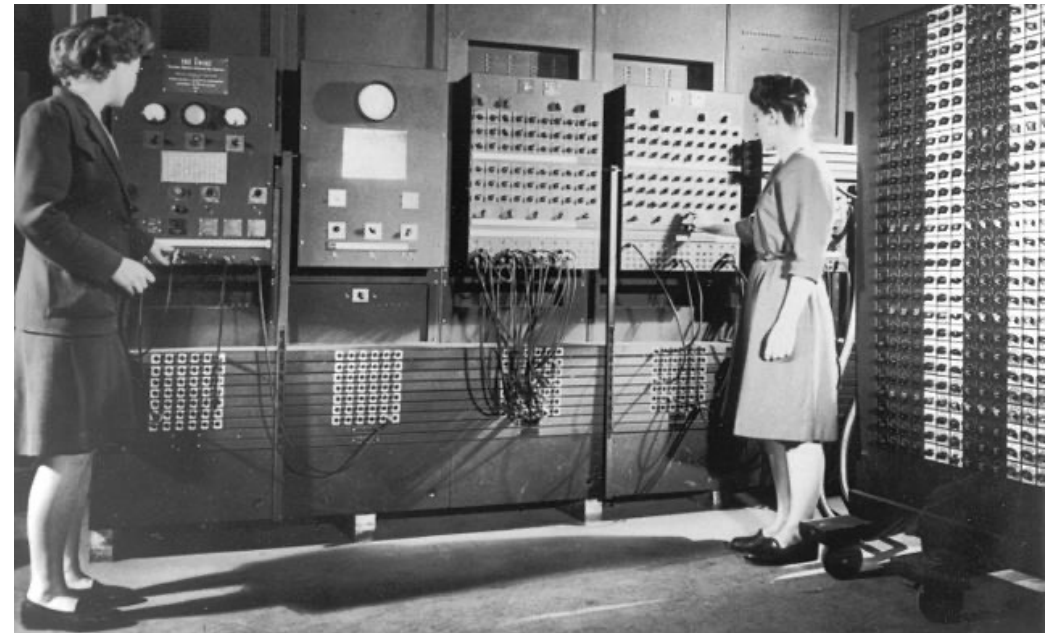
# The User Interface (UI) . . .

---

- ❑ Is the bridge facilitating interaction and communication between humans and computers
- ❑ Is one of the most important parts of a program because it determines how easily the user can make the program do the desired action
- ❑ Encompasses a wide range of elements, e.g., graphical elements, input methods, and navigation
- ❑ Is designed to make computing more accessible and intuitive

# History, Evolution and Types

- ❑ Early computers used switches and punch cards for input, which required specialized knowledge and training
- ❑ Operators had to manually configure these hardware components to set the computer's initial conditions



By Unidentified U.S. Army photographer - Image from Historic Computer Images, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=26253297>

# Command Line Interfaces (CLI)

---

- ❑ Command Line Interfaces (CLI) represented a significant shift from early interfaces by introducing text-based commands and scripts
- ❑ Users interacted with computers by typing text commands
- ❑ This required users to learn specific commands and syntax, making it less user-friendly

```

C:\> Get-ChildItem 'MediaCenter:\Music' -rec |
>> where < -not $_.PSIsContainer -and $_.Extension -ma
>> Measure-Object -property length -sum -min -max -ave
>>

Count      : 1307
Average    : 5491276.09563887
Sum        : 7177097857
Maximum    : 22905267
Minimum    : 3235
Property   : Length

PS C:\> Get-WmiObject CIM_BIOSElement | select biosv*, man*, s

BIOSVersion : <TOSCPL - 6040000, Ver 1.00PARTBL>
Manufacturer : TOSHIBA
SerialNumber : M821116H

PS C:\> <[wmiSearcher]@'
>> SELECT * FROM CIM_Job
>> WHERE Priority > 1
>> '().get() | Format-Custom
>>

class ManagementObject#root\cimv2\Win32_PrintJob
{
    Document = Monad Manifesto - Public
    JobId = 6
    JobStatus =
    Owner = User
    Priority = 42
    Size = 1027088
    Name = Epson Stylus COLOR 740 ESC/P 2, 6
}

PS C:\> $rssUrl = 'http://blogs.msdn.com/powershell/rss.aspx'
PS C:\> $blog = [xml](new-object System.Net.Webclient).Download
PS C:\> $blog.rss.channel.item | select title -first 3

title
-----
MMS: What's Coming In PowerShell V2
PowerShell Presence at MMS
MMS Talk: System Center Foundation Technologies

PS C:\> $host.version.ToString().Insert(0, 'Windows PowerShell
Windows PowerShell: 1.0.0.0
PS C:\>

```

# Command-line interface prompt

- ❑ A text-based program used for viewing, handling, and manipulating data and accepts typed command on a line has a command-line interface.
- ❑ The program displays a symbol or set of symbols indicating that the computer is waiting for a command to be typed. This is called a **prompt**.



# Graphical User Interface (GUI) Design

# Graphical User Interfaces (GUI)

---

- ❑ Developed in the 1970s and 1980s
- ❑ Computing more accessible with a visually intuitive and interactive environment
- ❑ Introduced visual elements
- ❑ Graphical elements: icons, buttons, windows, etc.
- ❑ Mouse-Based Navigation
- ❑ Multitasking
- ❑ WYSIWYG (What You See Is What You Get)
- ❑ Accessibility and Inclusivity

# Graphical User Interface (GUI)

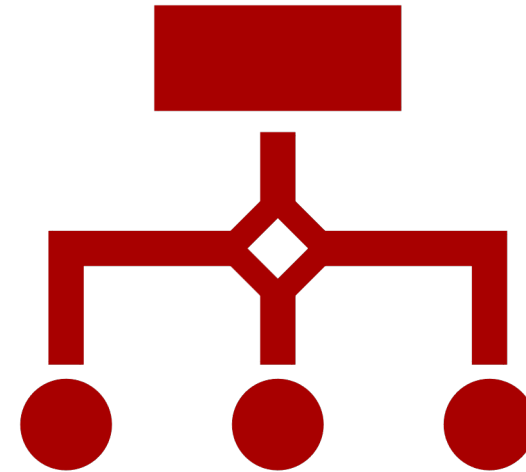
---

- An interface program that allows users to interact with a computer through its graphic display by using a mouse, track pad, or other peripheral to point and click on graphics or icons.
- A response to the problem of inefficient **usability** in text-based command-line interfaces for non-technical users
- The standard of **user-centered** design in software application programming
- Provides users the capability to **intuitively** operate computers and various devices



# Menu-Driven Interfaces

- ❑ A type of user interface that relies on a hierarchical menu system to allow users to navigate and interact with an application
- ❑ Presents users with a list of options or commands organized in a menu structure
- ❑ A menu-driven interface is implemented within other interface types, text or graphical



# GUI, A response to the problem of inefficient usability

Text-based command-line interfaces (CLI) often had steep learning curves

CLI's require commands to be typed on a keyboard

In a GUI actions are performed through direct manipulation of graphical elements

The actions of a GUI do the same thing as commands typed on a keyboard

GUI and CLI  
programs do  
the same  
thing

A program stores data in variables which represent information in the real world

The value of these variables at any given moment when the program is running is called the program's **state**

A program's internal processes or interaction with a user or with its environment consists of actions or events such as accepting input or producing output, that may or may not cause it to change its state

# Things that have 'states' in the real world



Stop lights at an intersection



Waiting for service at the Secretary of State's office for driver's license services



An HVAC system in a house



Sorting produce and shipping to the correct store



Calculating purchase orders to maintain inventory

# A program represents 'states' and related information in the real world

---

- ❑ A running program represents the state of something in the real world
- ❑ The program contains data that represents that state
- ❑ When a program is stopped, if the data isn't preserved, the state will be lost
- ❑ If the state it was in is preserved and the program can be started and resume the state it was in before it was stopped, this is called **persistence**

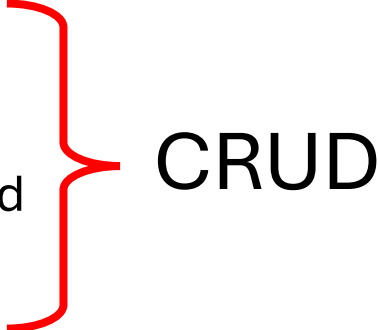
# Persistence and data structures

- ❑ Persistence is achieved by storing the state as data in a computer data storage device, e.g., a file or a database.
- ❑ Programs need to transfer data to and from storage devices
- ❑ When data is transferred to a program, the data is structured and held in the computer's memory (RAM)
- ❑ A **data structure** is a collection of data organized in some fashion
- ❑ The structure not only stores data **but also supports operations for accessing and manipulating the data**

# Data structures as classes

- ❑ A data structure is a collection of data organized in some fashion
- ❑ The structure not only stores data but also supports operations for accessing and manipulating the data
- ❑ In object-oriented programming, a data structure, also known as a container is an object that stores other objects, referred to as data or elements
- ❑ Defining a data structure is defining a class
- ❑ A data structure class should use data fields to store data and provide methods to support whatever operations need to be performed on the data
- ❑ Java provides several data structures commonly known as **Java Collections Framework**

# Collections, operations and CRUD

- ❑ The internal processes of a program, processes that result from interaction with a user or interaction with its environment cause the state of a program to change.
  - ❑ There are four basic operations that allow a program to change the data representing a state and cause the state to persist.
    - ❑ **Create**
    - ❑ **Read**
    - ❑ **Update, and**
    - ❑ **Delete**
- 
- CRUD**



# What is CRUD?

An acronym that stands for Create, Read, Update, and Delete

The four fundamental operations that can be performed on data

CRUD operations are essential for managing and manipulating data within a system

# Create, Read, Update, Delete

---

- ❑ **Create.** The creation of new data records or entries in a database or software system
- ❑ **Read**
  - ❑ Retrieving or viewing existing data records from a file or database
  - ❑ Allows users to access and view information without modifying it.
- ❑ **Update.** Modifying or altering existing data records
- ❑ **Delete.** Permanently remove data

# CRUD and the User Interface

---

- ❑ CRUD operations are closely related to user interfaces (UI) in software applications
- ❑ They dictate how users interact with and manipulate data within the application
- ❑ User interfaces are responsible for presenting options and controls to users, allowing them to perform CRUD operations seamlessly

# CRUD and the User Interface

---

- CRUD is also used to describe user interface capabilities that facilitate viewing, searching, and changing information using a computer.
- Data can be stored outside a program or the computer's memory.
- The data contained in the storage is both readable and updatable.
- Before the data can be read or updated, it must be created
- At some point in time, some or all data may need to be deleted.

# User Interface Design

---

Creating a User Interface (UI) involves designing and building the visual components of a software application that allow users to interact with the program through an interface.

# User Interface Design

---

- ❑ A user interface allows a user to perform CRUD operations and related sub-operations on data
- ❑ User interfaces need to be designed
- ❑ User interface (UI) design is the design of user interfaces for software with the focus on maximizing ease of use
- ❑ The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals

# User Interface Design Process

1. **Plan:** Before you start coding, sketch a rough layout of your UI, including the placement of buttons, labels, input fields, and other elements. How will users interact with your application?
2. **Design:** Use a tools to design the user interface. Add widgets such as buttons, text fields, labels, checkboxes, and dropdown menus to your GUI
  - Consider accessibility features for users with disabilities
3. **Define Functionality:** Write code to define the functionality of your GUI. This includes event handling (e.g., what happens when a button is clicked) and data processing.
4. **Test and Debug:** - Test your UI thoroughly to ensure that it works as expected. Correct any issues that arise
5. **Optimize and Refine:** Optimize the performance of your UI and refine the design based on user feedback and usability testing.

# Planning the UI

1. **User Requirements Analysis:** Start by understanding the needs and preferences of your target users. What are their goals and tasks when using your application? Conduct user interviews or surveys to gather valuable insights.
2. **Use Cases and Workflows:** Create use cases and workflow diagrams to visualize how users will navigate through your application and perform tasks. This helps in identifying the key features and components needed.
3. **Information Architecture:** Organize the content and functionality of your application logically. Determine how data will be structured and presented to users.
3. **Wireframing and Mockups:** Create low-fidelity wireframes or mockups to sketch the layout and structure of your GUI. Tools like Balsamiq, Sketch, or Adobe XD can be useful for this.
4. **Prototyping:** Develop interactive prototypes to simulate the user experience. Prototyping tools like Figma, Adobe XD, or InVision can help you create clickable prototypes.
5. **Usability Testing:** Conduct usability testing with potential users to gather feedback on your wireframes or prototypes. Use this feedback to refine your design.



# Wireframing and mockup

Wireframing and mockup are essential steps in the GUI design process

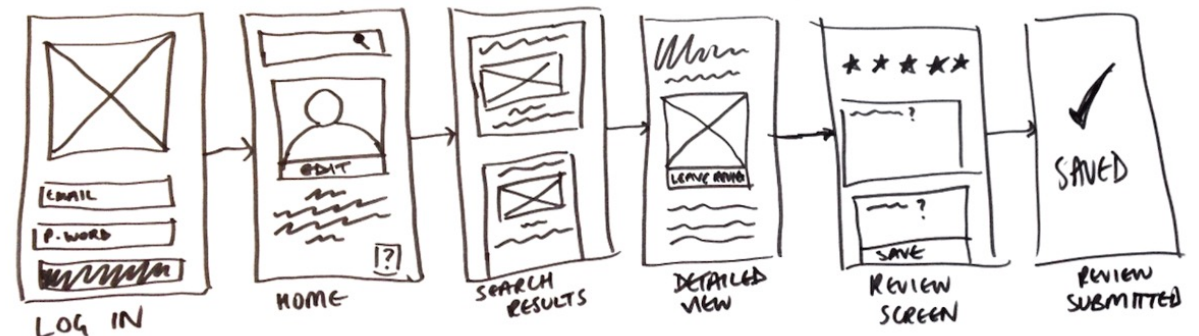
Allow you to create visual representations of your interface's layout, structure, and functionality before diving into full-fledged design and development

# What Is Wireframing?

**wireframe** — a simple, low-fidelity diagrams or sketches that outline the basic structure and layout of your GUI without focusing on visual detail

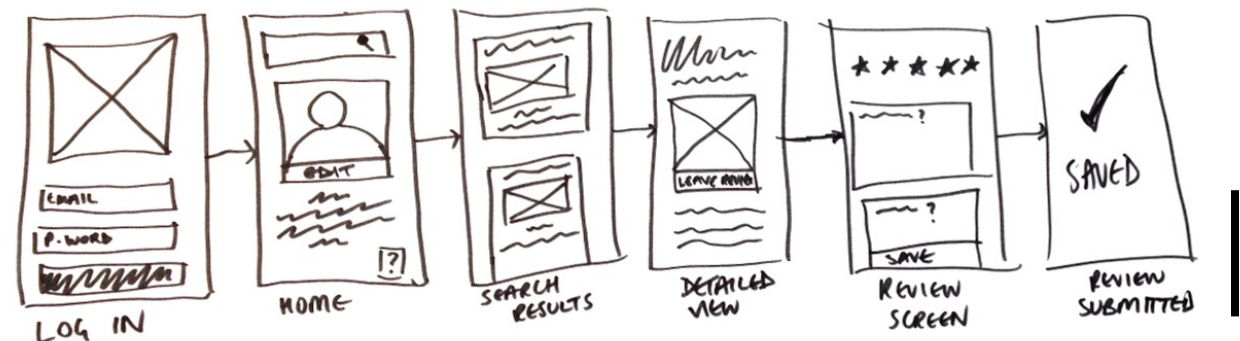
- ❑ Helps define placement of components and flow of information
- ❑ Presents proposed functions, structure and content
- ❑ It is used to express how users will interact with a program
- ❑ The blueprint of the GUI

- ❑ Can be created in black and white, and
- ❑ Can begin with hand drawings



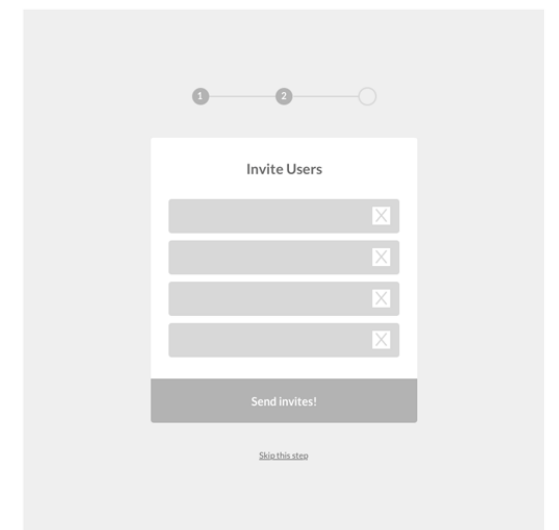
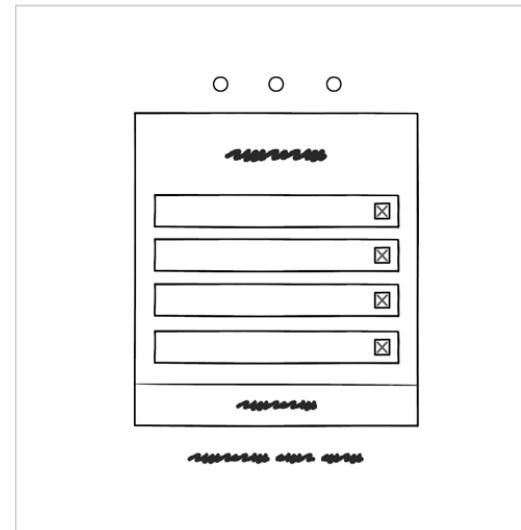
# Creating a wireframe

- Identify the data involved
- Identify how the GUI will display the data and/or allow the data to be processed when it is created, read, updated and deleted
- Create a wireframe for each different way the data may be displayed or processed

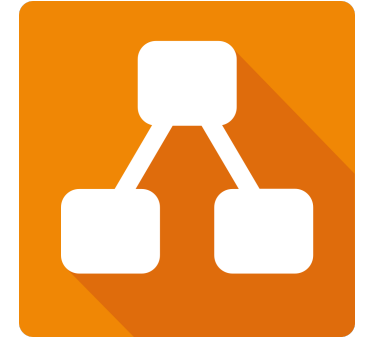


# Wireframe tools

- Wireframing can also be created using software
- Elements shown in wireframes include various controls, e.g., input fields, search fields, labels, buttons and text areas, etc.



# Diagrams.net



**diagrams.net** is a free and open-source cross-platform graph drawing software

- can be used to create diagrams such as flowcharts
- available as online as cross-browser web app, and as offline desktop application for Linux, macOS, and Window
- integrates with cloud services for storage including Dropbox, OneDrive, Google Drive, GitHub, and GitLab.com

# User Interface example: music player

---

- The basic storage unit is an individual song
- A playlist will consist of one or more songs

Music player must allow the user to:

- Create a new playlist
- Read (retrieve) and display stored playlist
- Update a playlist
  - Add a song to the playlist
  - Delete a song from the playlist
- Persistence: store the playlist

# Music player GUI - Identify the data involved

The basic storage unit is an individual song

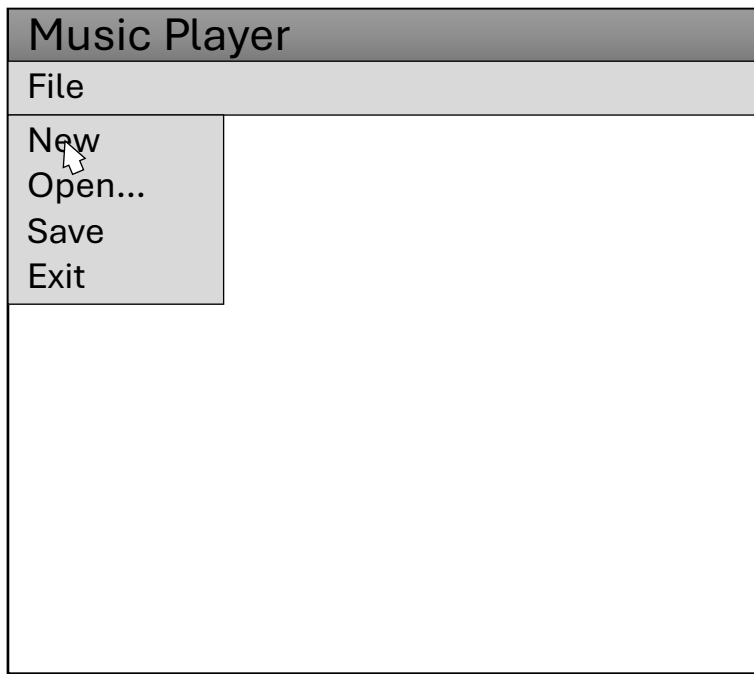
- Artist
- Song Title

A playlist will consist of one or more songs



Playlist	
Artist	Song Title
Beyonce	Break My Soul
Harry Styles	As It Was
Kate Bush	Running Up That Hill
DJ Khaled Ft Drake & Lil Baby	Staying Alive
Steve Lacy	Bad Habit
Future Ft Drake & Tems	Wait For U
Nicky Youre & dazy	Sunroof
Jack Harlow	First Class
benny blanco, BTS & Snoop Dogg	Bad Decisions

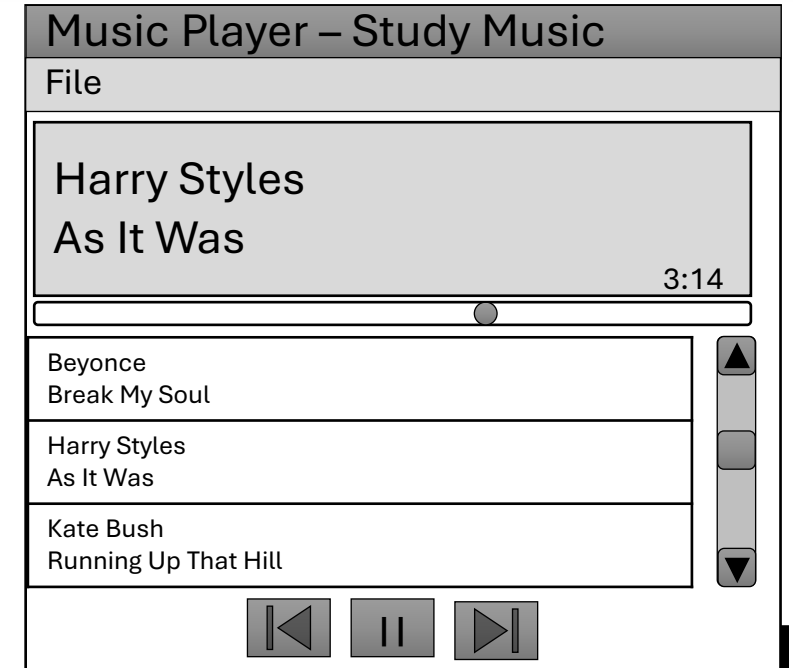
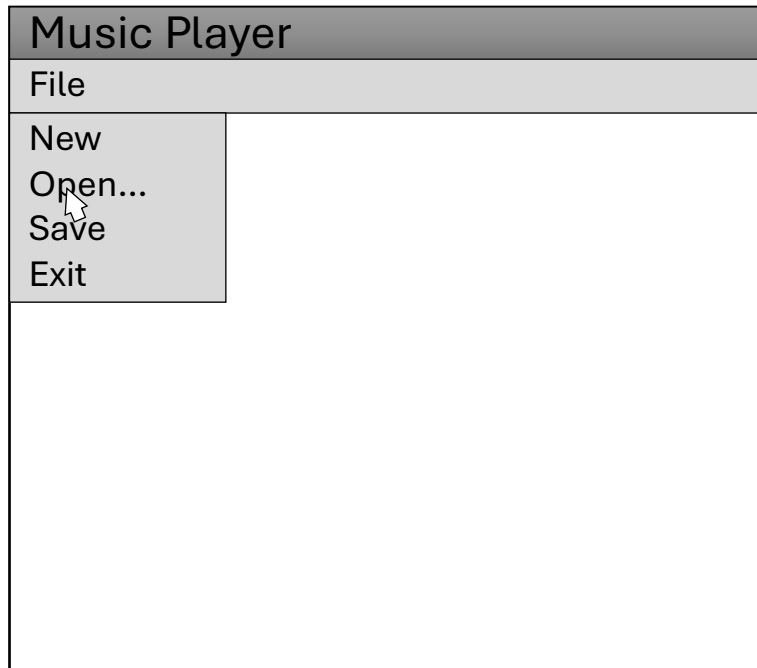
# Music player GUI - Create a new playlist



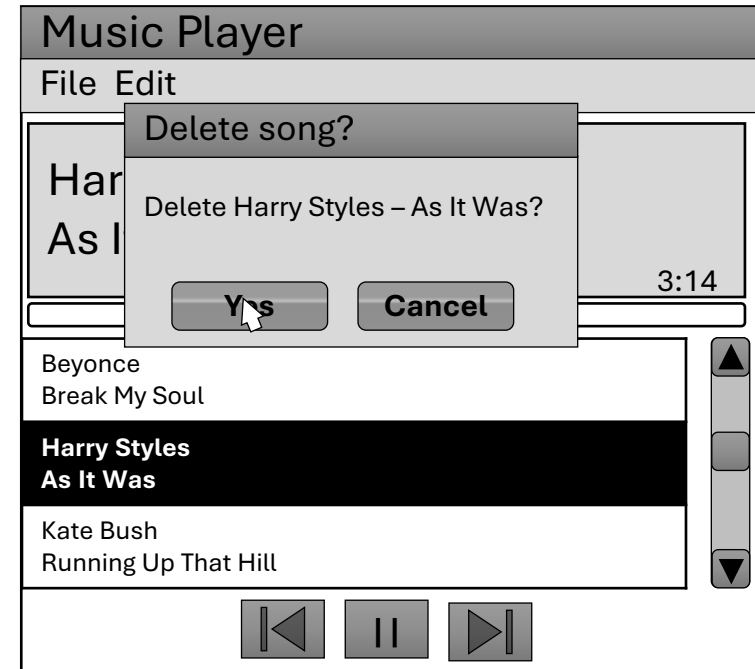
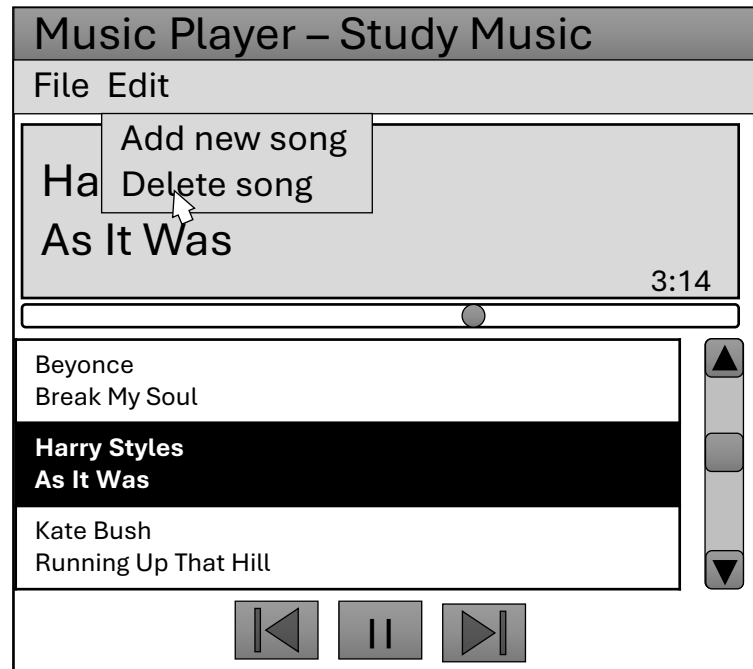


# Music player GUI

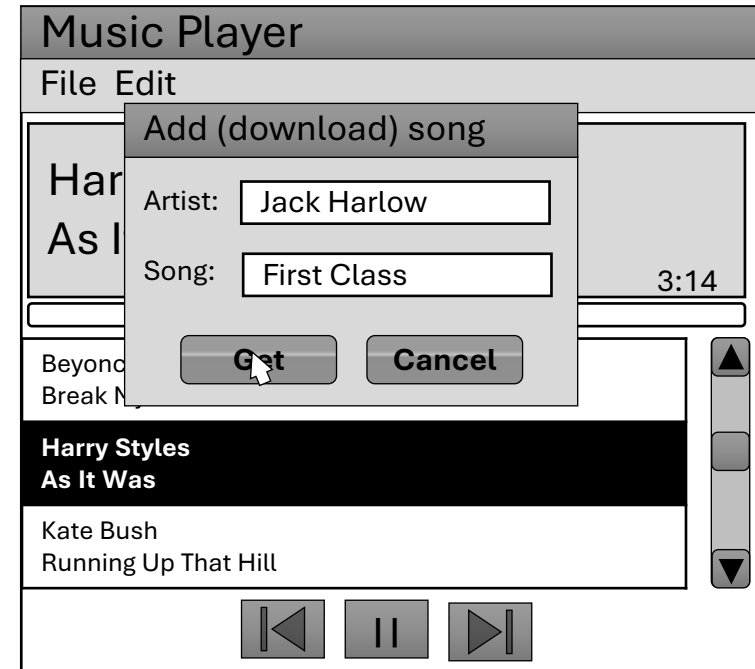
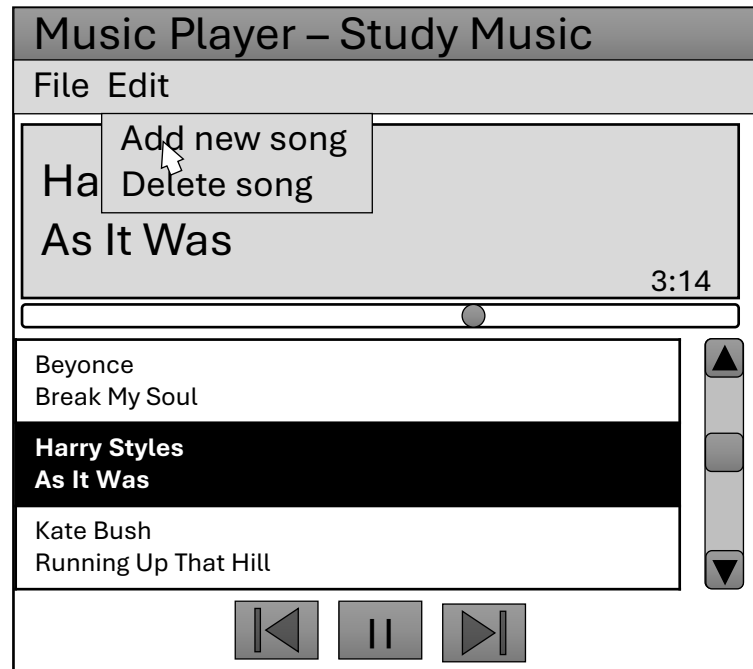
## Read (retrieve) and display stored playlist



# Music player GUI - Uppdate a playlist



# Music player GUI - Uppdate a playlist



# Music player GUI – Persistence: store the playlist

