## horizontal line



Project Proposal: Visual Note-Taking System with Classes and Structures

05.03.2025

**─**

Dmytro Gozha

**Course:** CS2

**Project Name:** Structured Canvas Notes

**Las Positas Community College**

# Overview

This project is a graphical note-taking application where users can create and connect data blocks using the concepts of **classes** and **structures**, inspired by object-oriented design. The user can create variables (like strings, numbers, dates, images, or long notes), organize them visually, and establish relationships between them using arrows. The canvas supports flexible modeling for personal, academic, or financial use.

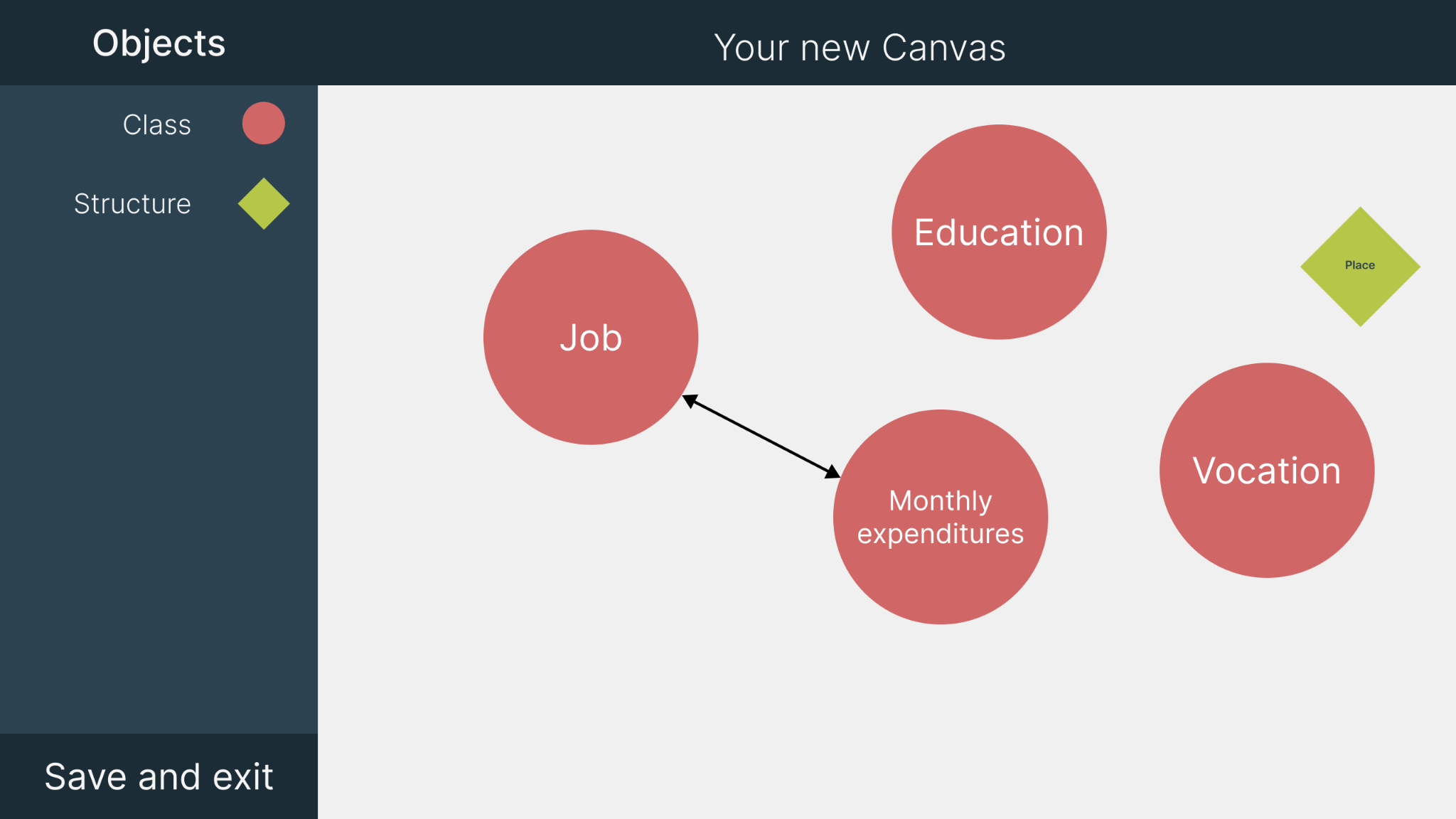
# Goals

1. Allow users to visually manage complex information.
2. Introduce beginner-friendly object-based thinking using structures and classes.
3. Enable links between data blocks for automation and logical grouping.
4. Create an intuitive and interactive canvas workspace for note creation.

# Core Components

#### **Canvas**

* A main working area where all classes and structures are placed.
* Zoomable and pannable.
* Right-click to open the **“Create New”** menu (to add Class or Structure).
* Option to switch between canvases using tabs.

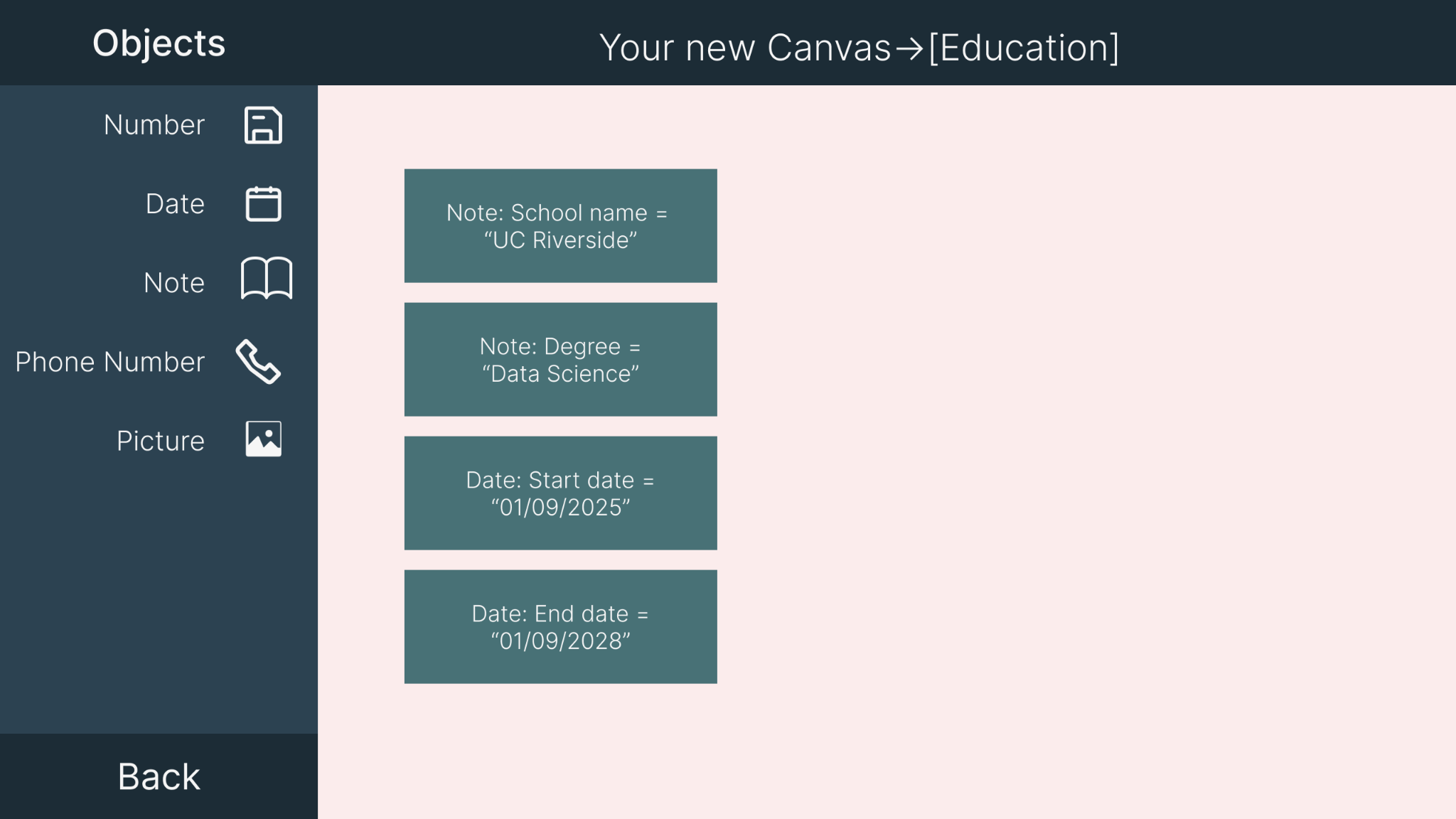


#### **2. Class**

* A container for real data and instances.
* Supports adding variables of the following types:  
  + String
  + Number (including long/float/double types)
  + Date
  + Phone Number
  + Image
  + Note (multi-line paragraphs)
* Each class can have a name and position on the canvas.
* Supports **linking** to Structures via arrows.

##### **Buttons Inside Class Block**

* + Add Variable (with dropdown menu for variable type)
* 🖊️ Edit (edit class name or variables)
* 🗑️ Delete
* ↔ Link Structure (connects a Structure to the class)

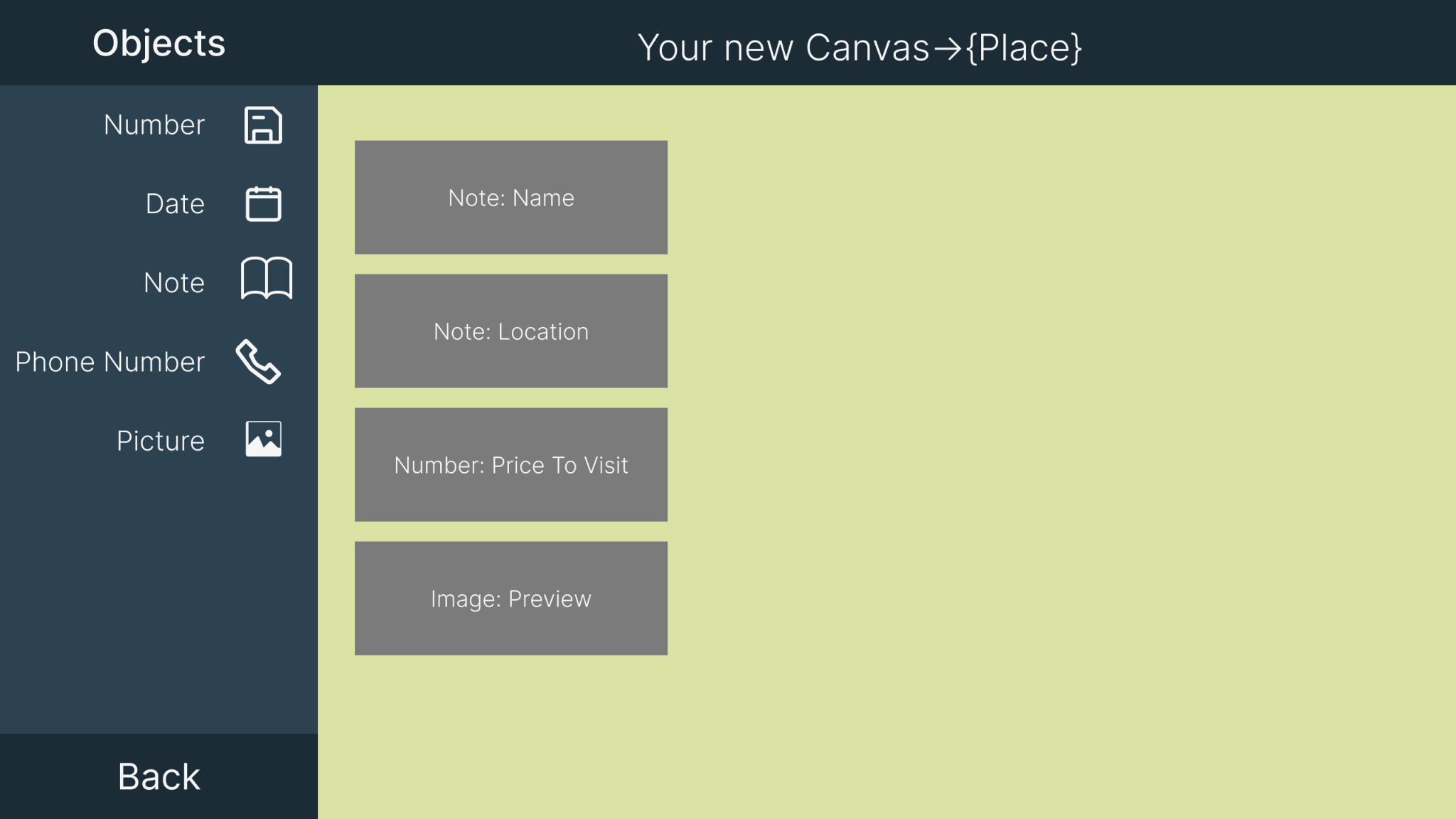


#### **3. Structure**

* A reusable blueprint for variable templates.
* Does not hold actual data, only variable definitions.
* Examples: Place, Education, Person, Contact, etc.
* Can be linked into multiple Classes.
* Once linked, Structure variables appear inside the Class as editable fields.

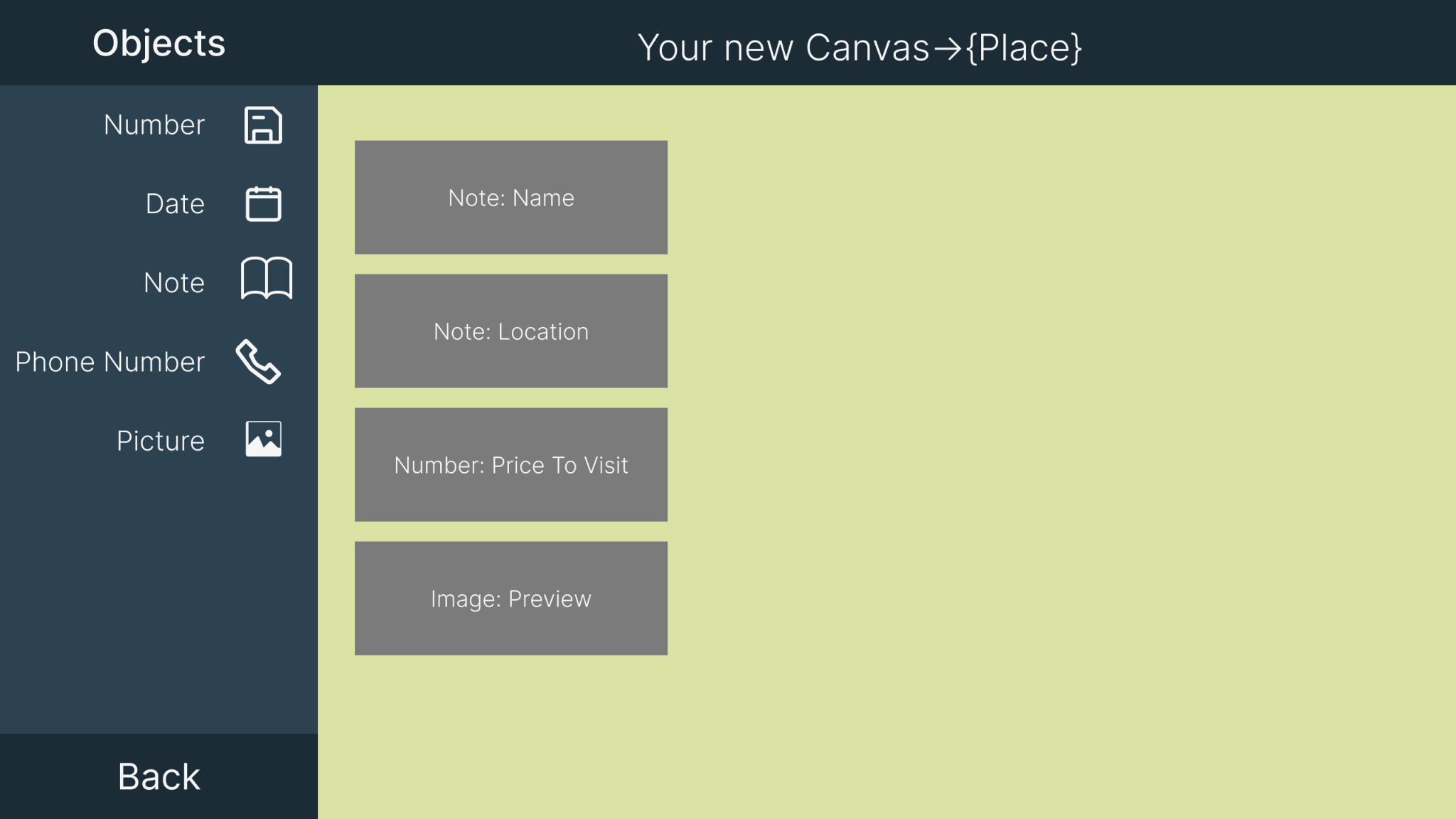
##### **Buttons Inside Structure Block**

* + Add Field (string, number, etc.)
* 🖊️ Edit
* 🗑️ Delete

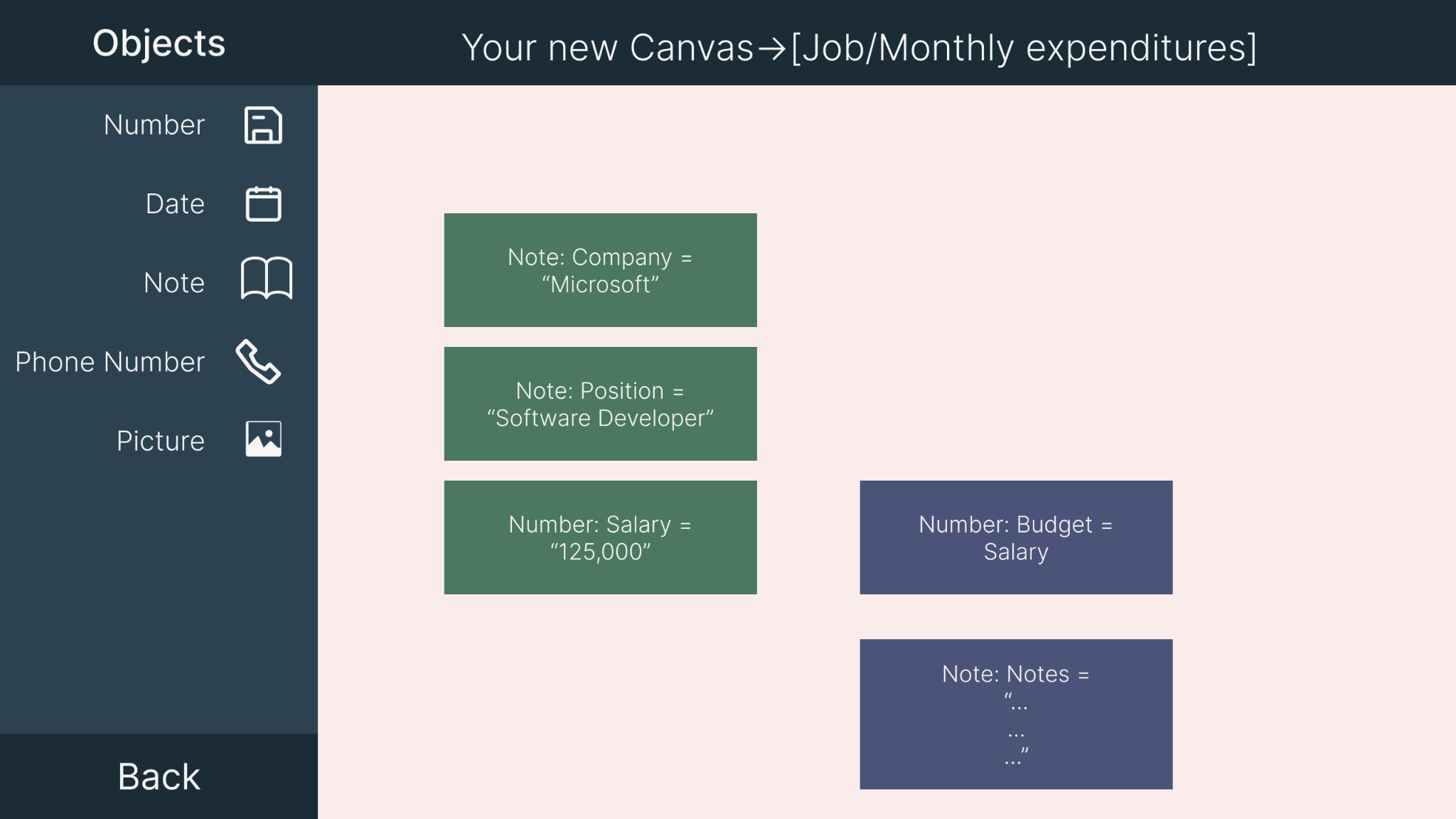


#### **4. Linking**

* Dragging an arrow from a Class to a Structure creates a link.



* Dragging from one Class to another allows linking variable values (ex: Salary → Budget).



* Linked variables auto-update if the source changes.

#### **5. Side Toolbar**

* 🞧 Create New
* 🔍 Search
* 🧲 Snap to Grid
* 💾 Save / Export
* 🗂 Canvas List
* 🧽 Clear Canvas

# Data Types Supported

| **Type** | **Description** |
| --- | --- |
| String | Single line or word-based text |
| Number | Includes integers, floats, doubles |
| Date | Calendar date selection |
| Phone Number | Formatted phone input |
| Image | Upload or paste image |
| Note | Multiline text block |

# Specifications

The project will be developed using C++ (language of the course), and rendered in a graphical interface, most likely using a GUI framework like **Windows Forms** or **SFML**. It will be a desktop-based standalone application.

| **Feature** | **Specification Details** |
| --- | --- |
| Language | C++ (CS2 requirement) |
| GUI Library | SFML / ImGui / WinAPI (TBD) |
| Canvas | 2D coordinate grid, panning, drag-and-drop |
| Data Storage | In-memory (optional: file save/load as JSON or custom format) |
| Variables Supported | String, Number (int/double), Date, Phone, Image, Note |
| Class Capabilities | Store data, link to structures, display variable blocks |
| Structure Capabilities | Define reusable variable templates |
| Linking Logic | One-to-many arrows, update propagation, value synchronization |
| UI Elements | Buttons, dropdowns, arrows, zoom, right-click context menus |
| Performance Goal | Smooth handling of 50+ nodes on canvas with responsive UI |
| Platform | Windows (initial target) |
| Export Options (Optional) | JSON / PDF export of canvas data |
| File Save / Load (Optional) | Serialize entire canvas into a file and reload |

# Milestones

| **Week** | **Goals** |
| --- | --- |
| **Week 1**(Current) | ✅ **Design & Setup**Finalize idea & structureCreate Figma mockups  Set up GUI window with chosen framework |
| **Week 2** | 🔧 **Core Development**  Render canvas and place objects  Implement class and structure logic  Add variable system and editing  Arrow connections between objects |
| **Week 3** | 🎯 **Final Touches & Examples**  Polish UI & fix bugs  Build examples: Job ↔ Budget, Vacation + Places  Take screenshots  Prepare for presentation (Google Slides or demo) |

# Conclusion

This project combines computer science concepts (classes, variables, data types) with real-world organization (budgeting, planning, education tracking). It enhances learning through interaction and can be extended into a full productivity tool. Each part of the app supports logic, connection, and clean visual representation of structured data.