

# **CASE TOOL FOR STRUCTURED DESIGN**

**FINAL PROJECT WORK - SOFTWARE ENGINEERING LAB CS3074**

Submitted By-

Devidutta Nayak - 119cs0144

Utsab Ojha - 119cs0145

Paresh kumar Pagal - 119cs0147

Aditya Narayan Panda - 119cs0148

# Contents

PAGE NO

1. <u>SOFTWARE REQUIREMENT SPECIFICATIONS .....</u>	<u>1</u>
2. <u>UML DIAGRAMS.....</u>	<u>6</u>
3. <u>CODE .....</u>	<u>10</u>
4. <u>CODE INTERFACE .....</u>	<u>26</u>
5. <u>CONTRIBUTIONS .....</u>	<u>28</u>

## **INTRODUCTION**

This Computer Aided Software Engineering tool software is to be developed to perform the Structured design in the process of developing a Software. This tool helps to automate various activities associated structured software design so that user can select various design components and combine it to get the desired structured design. This structure design will help the user to visualize complex things easily for further development of software

### **1. Software Requirement Specification**

#### **1.1 Purpose:**

- 1.1.1 Main aim in developing structured design editor is to provide a user friendly interface mainly for the novice users as well as expert users to integrate different kinds of structured design components to design a software.
- 1.1.2 The intended audience is any person/designer who wants to design the structure of a software.

#### **1.2 Scope:**

- 1.2.1 The product is titled “Case tool for structured design”.
- 1.2.2 The product will perform the following tasks
- 1.2.3 The case tool should support a graphical user interface and the following features
- 1.2.4 The user should able to draw modules, control arrows, data flow arrows with their corresponding data name, symbol for library modules etc
- 1.2.5 Diagrams is arranged in neat hierarchical levels
- 1.2.6 User should able to modify his design
- 1.2.7 Clicking on a module should be able to show its internal organization
- 1.2.8 User should be able to save his design and also be able to load previously created design

#### **1.3 Overview:**

- 1.3.1 The SRS contains an analysis of the requirements necessary to help easy design.
- 1.3.2 The overall description provides interface requirements for the Case Tool for structured design, graphical interfaces, software interfaces, relationship between classes used, product functions etc.
- 1.3.3 This system provides the detail structure of all the components, integrating process, dismantling of the components.

1.3.4 Succeeding pages illustrate the characteristics of typical naïve users accessing the system along with legal and functional constraints enforced that affect Case Tool for Structured design in any fashion.

#### 1.4 The overall description:

##### 1.4.1 Product perspective

###### 1.4.1.1 Hardware Interface:

1.4.1.1.1 The system must interface with the standard output device, keyboard and mouse to interact with this software.

###### 1.4.1.2 Software Interface:

1.4.1.2.1 Front-End: HTML,CSS,BOOTSTRAP, JAVASCRIPT

###### 1.4.1.3 Memory Constraints

1.4.1.3.1 No specific constraints on memory

###### 1.4.1.4 User Characteristics

1.4.1.4.1 The intended users of this software need not have specific knowledge as to what is the internal operation of the system. Thus the end user is at a high level of abstraction that allows easier, faster operation and reduces the knowledge requirement of end user .

1.4.1.4.2 The Product is absolutely user friendly, so the intended users can be the naïve users.

1.4.1.4.3 The product does not expect the user to possess any technical background. Any person who knows to use the mouse and the keyboard can successfully use this product

#### 1.5 Specific Requirements

##### 1.5.1 **Functional Requirements**

###### 1.5.1.1 **File**

1.5.1.1.1 New: This option will give a new drawing pad to draw a new structured design. In case a design is opened already, then a confirmation to Save/Cancel will be asked.

1.5.1.1.2 Open: Whenever user will click this option, a new file browser window will open and user can select previously saved circuit.

- 1.5.1.1.3 Save: By clicking this option, a new file browser window will be opened, where user can save the current drawn design in XML/JSON format which can be used in future.
- 1.5.1.1.4 Save As: This option is for saving an existing design with a new name
- 1.5.1.1.5 Export: Here the user can save the drawn circuit in some special formats like JPEG, JPG, PDF etc. These formats are not modifiable. So, these exported files can't be used in future for modifying the circuit. But sometime, these kinds of formats are needed for different purpose. So, this useful feature needs to be incorporated.
- 1.5.1.1.6 Exit: The application will be closed. An extra checking option will be there for the unsaved projects after the click the Exit menu item.

### **1.5.1.2 Edit**

- 1.5.1.2.1 Cut: This option will remove the selected components from the drawing/designing area and copy them in the clipboard.
- 1.5.1.2.2 Copy: This option will copy selected components of the design in the clipboard.
- 1.5.1.2.3 Paste: The copied or cut portions will be placed in another position in the drawing board where the user wants.
- 1.5.1.2.4 Select Component: User can select the components which he/she wants to include in the design.
- 1.5.1.2.5 Remove component: User can remove the desired component from the drawing board.
- 1.5.1.2.6 Reset Screen: This function is used to clear the whole drawing board.
- 1.5.1.2.7 Resize: Any desired component can be resized.
- 1.5.1.2.8 Update Label: Labels of the components can be changed by user's choice.
- 1.5.1.2.9 Fill Color: Color to the components can be applied to make the design to look attractive.
- 1.5.1.2.10 Select: User can select certain components in the design drawn.

- 1.5.1.2.11 Select All: The entire design can be selected by this menu item.
- 1.5.1.2.12 Select A Block: User can select a block (block consists of multiple components together) at a time and can perform necessary edit-work like copy, paste, cut, move etc.
- 1.5.1.2.13 Move : User can select any component and move it around the whole drawing pad. Along with the design, the connections with the certain components will also be updated.
- 1.5.1.2.14 Undo: The state, before the last action will be retrieved. This is a standard 'undo' action like all other application.
- 1.5.1.2.15 Redo: The state from where UNDO action was performed can be retrieved using REDO option. This is also a standard action.

### 1.5.1.3 View:

- 1.5.1.3.1 Full Screen: The drawing pad will occupy the full screen. This is for better viewing experience.
- 1.5.1.3.2 Grid: It will show grid in the drawing pad. User can set the size of the grid as per their choice. Three Types of grids are possible (Small, Medium, Large). User can also go for no grid.

### 1.5.1.4 Tools:

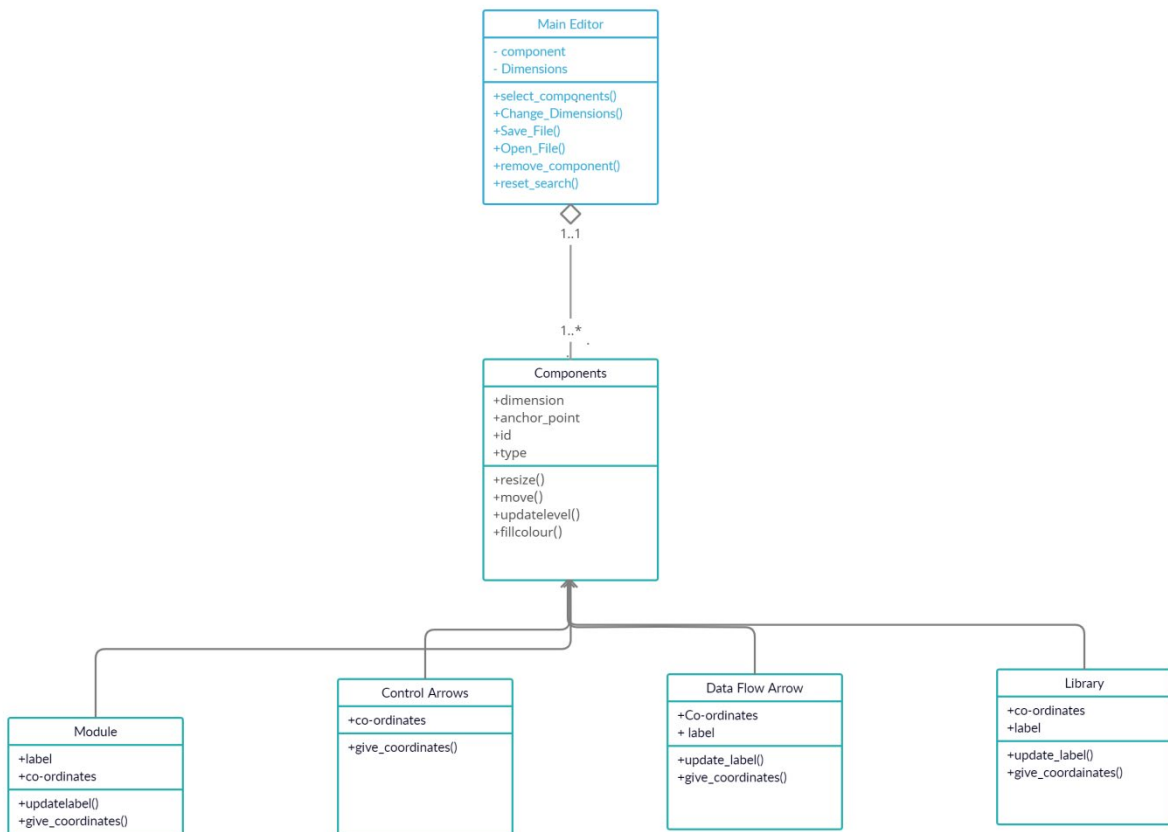
- 1.5.1.4.1 Install new component package: If the user wants to add some new design components then user can select this option. Again a File browser will open to select the package file (zip). The package will contain the design components and their functionality within it. It will provide the necessary information so that the application can add the component into its functionality.
- 1.5.1.4.2 Uninstall component package: if the user wants to remove some installed packages then user can select this option. After selecting this, new window will open where user can

select among all the installed packages which user wants to remove. All the information, related to the component will also be removed

#### **1.5.2 Non Functional Requirements:**

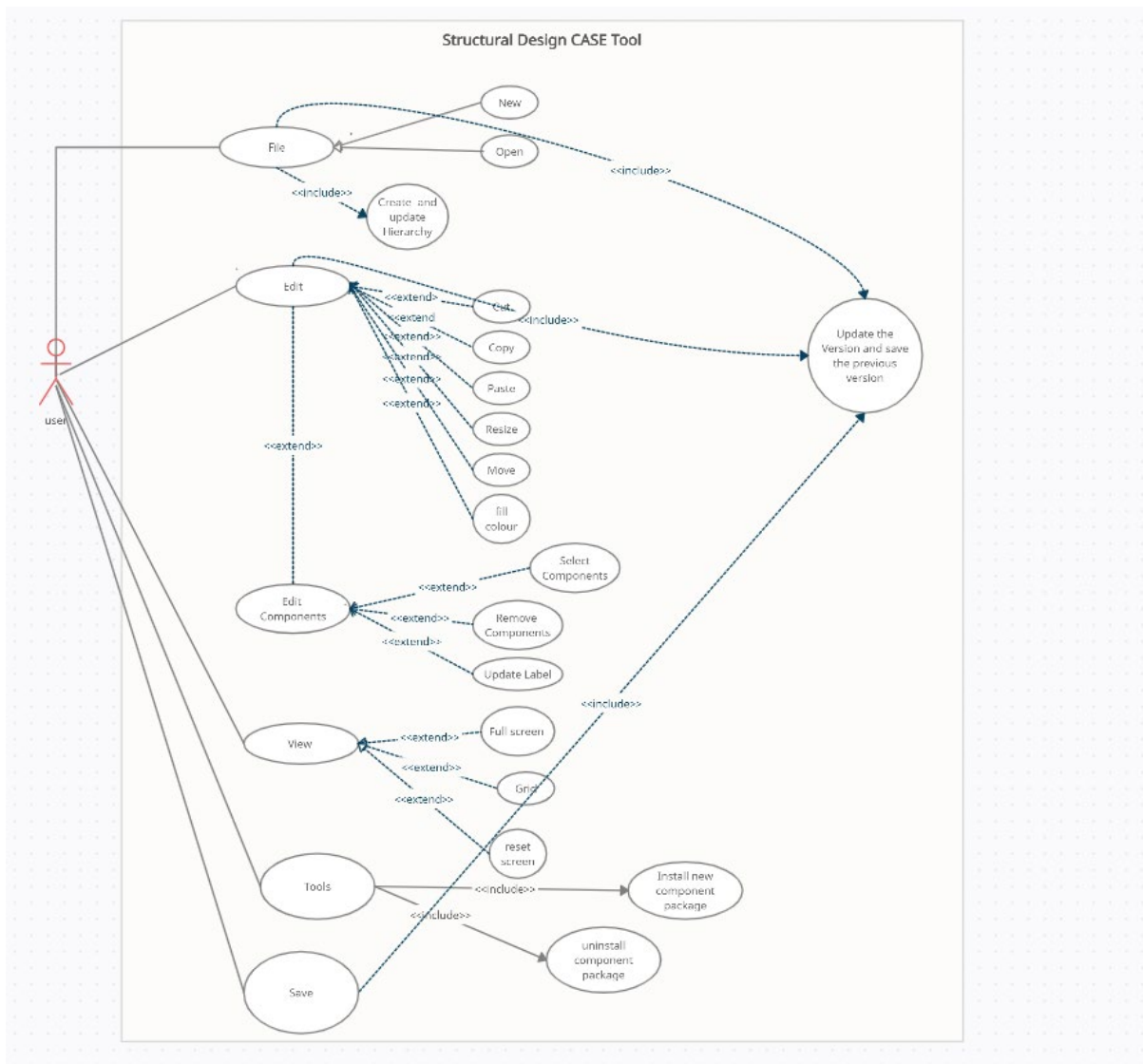
- 1.5.2.1 The software should be reliable i.e it should be available to user 24\*7 with minimum down time.
- 1.5.2.2 There shouldn't be any lag while selecting the component to use.
- 1.5.2.3 If user wants to add some new design components to the design, then he/she should have a way to do it. So the software should be scalable.
- 1.5.2.4 The design document should be saved in some definite time interval so that in case the software crashes, the user would be able to retrieve the design document.
- 1.5.2.5 The user interface should be user friendly. Users from novice to Expert skill level should be able to use it with ease.

## CLASS DIAGRAM

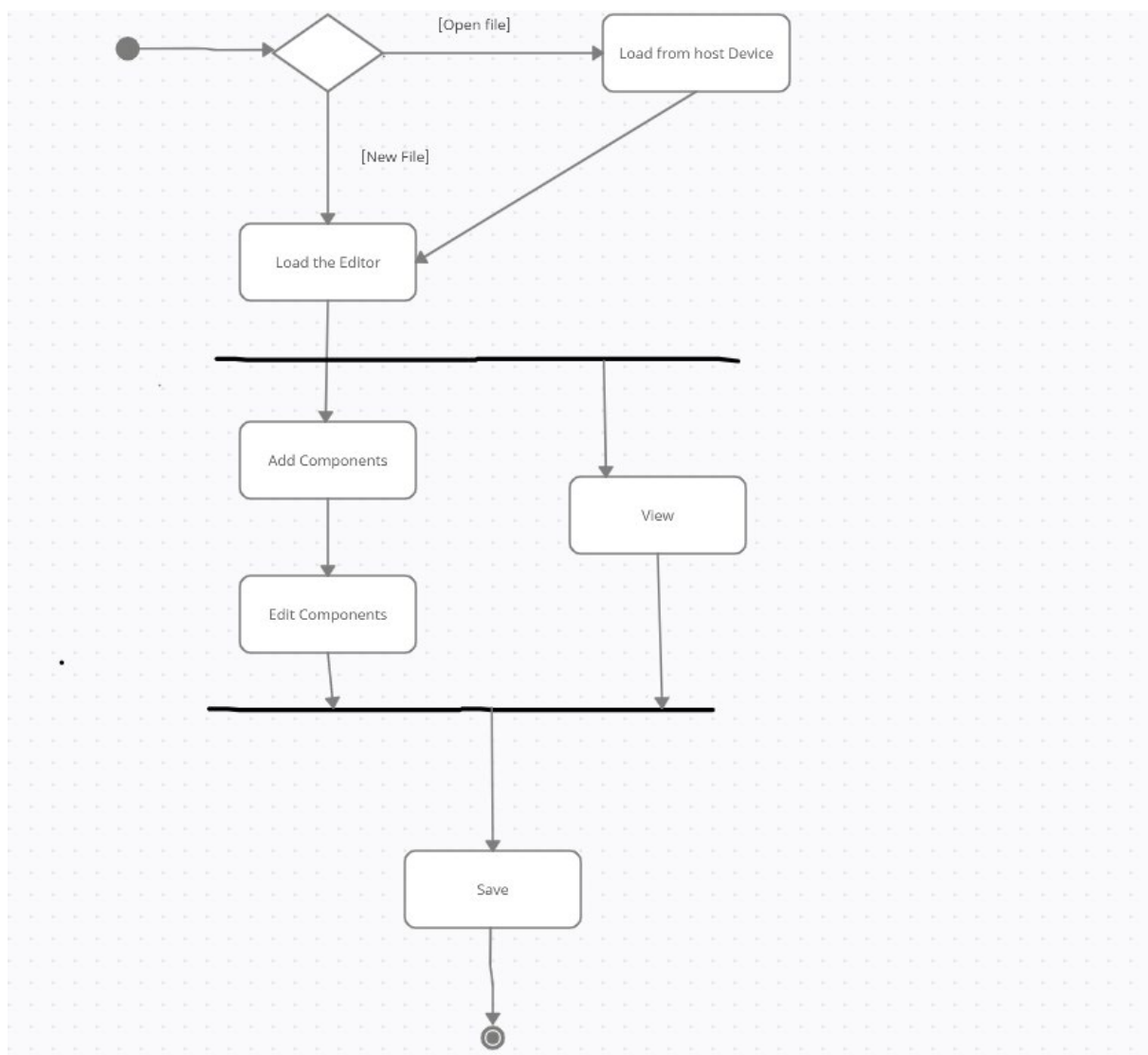




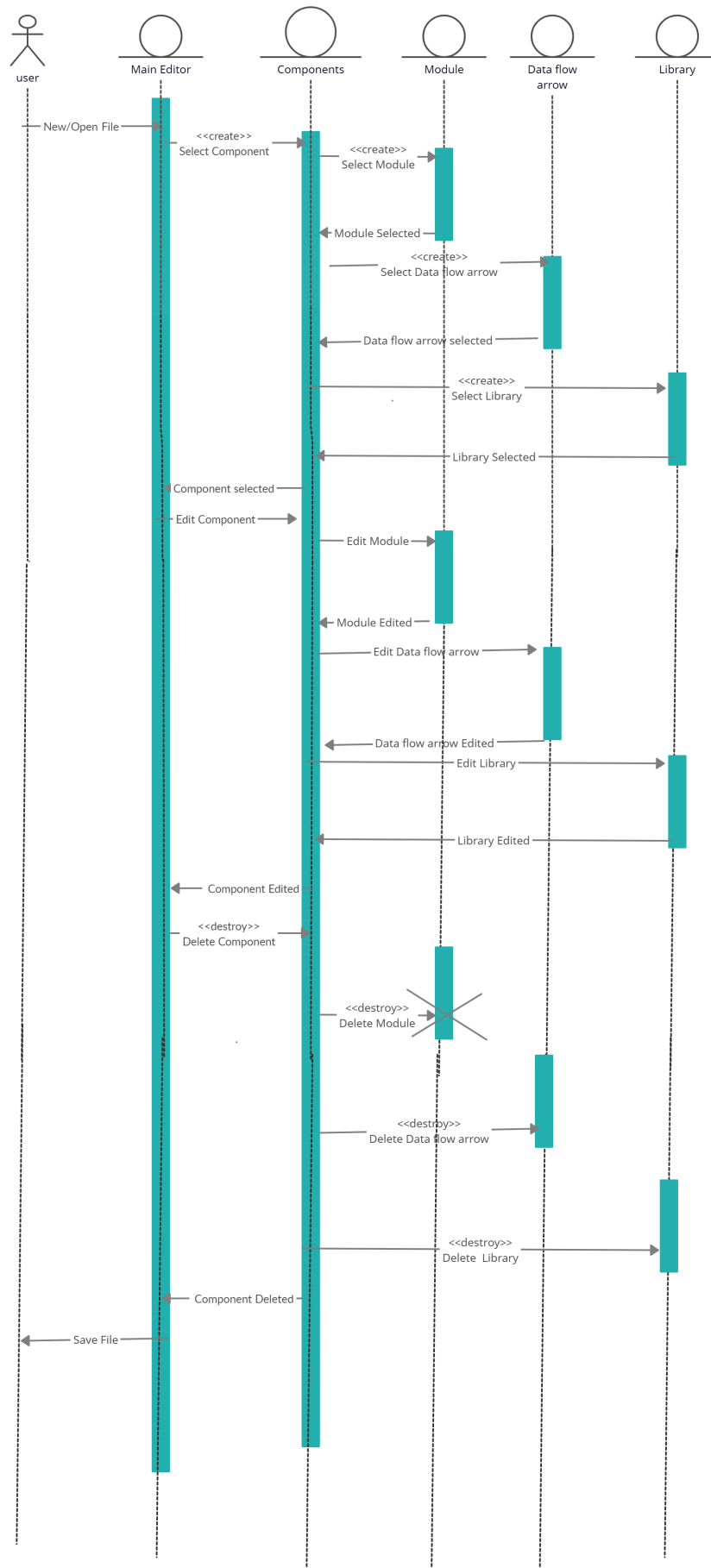
## USE CASE DIAGRAM



## ACTIVITY DIAGRAM



## **SEQUENCE DIAGRAM**



## Code

HTML Code:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <!-- <meta http-equiv="X-UA-Compatible" content="IE=edge" -->
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>CASE Tool | Draw diagrams with ease</title>
    <link rel="icon" href="images/logo.png" type="image/icon type" />
    <link
      rel="stylesheet"
      type="text/css"
      href="https://cdnjs.cloudflare.com/ajax/libs/jointjs/3.4.4/joint.css"
    />
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
    <link
      href="https://fonts.googleapis.com/css2?family=Cairo:wght@700&display=swap"
      rel="stylesheet"
    />
    <!-- CSS only -->
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy12QvZ6jIW3"
      crossorigin="anonymous"
    />
    <link rel="stylesheet" type="text/css" href="style.css" />
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <div class="container-fluid">
        <a class="navbar-brand" href="#">CASE Tool </a>
        <button
          class="navbar-toggler"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#navbarScroll"
          aria-controls="navbarScroll"
          aria-expanded="false"
          aria-label="Toggle navigation"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarScroll">
          <div class="d-flex justify-content-center w-100">
            <span
              class="badge bg-info text-dark display-2 file-name"
              id="file-name"
            >
            </span>
            <span class="badge" id="rename-file" style="cursor: pointer">
              <svg

```

```

        xmlns="http://www.w3.org/2000/svg"
        height="18"
        width="18"
        viewBox="0 0 20 20"
        fill="currentColor"
    >
        <path
            d="M17.414 2.586a2 2 0 00-2.828 0L7 10.172V13h2.828l7.586-7.586a2 2 0
000-2.828z"
        />
        <path
            fill-rule="evenodd"
            d="M2 6a2 2 0 012-2h4a1 1 0 010 2H4v10h10v-4a1 1 0 112 0v4a2 2 0 01-2
2H4a2 2 0 01-2-2V6z"
            clip-rule="evenodd"
        />
    </svg>
</span>
</div>

<div class="d-flex" id="change-file-name">
    <ul
        class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll"
        style="--bs-scroll-height: 100px"
    >
        <li class="nav-item">
            <a class="nav-link p-0 m-0" id="loadbtn">
                <form
                    id="jsonFile"
                    name="jsonFile"
                    enctype="multipart/form-data"
                    method="post"
                >
                    <fieldset>
                        <input type="file" id="fileinput" style="display: none" />
                        <div class="input-group m-0 p-0">
                            <div class="input-group-append" style="cursor: pointer">
                                <span
                                    class="input-group-text bg-transparent border-0 text-white"
                                    id="btnLoad"
                                    onclick="loadFile();"
                                >
                                    <svg
                                        xmlns="http://www.w3.org/2000/svg"
                                        height="18"
                                        width="18"
                                        fill="none"
                                        viewBox="0 0 24 24"
                                        stroke="currentColor"
                                        stroke-width="2"
                                    >
                                        <path

```

```

            stroke-linecap="round"
            stroke-linejoin="round"
            d="M4 16v1a3 3 0 003 3h10a3 3 0 003-3v-1m-4-8l-4-4m0 0L8
8m4-4v12"
        />
    </svg>
    <span> &nbsp;</span>Load
</span>
</div>
</div>
</fieldset>
</form>
</a>
</li>
<li class="nav-item">
    <div class="input-group p-0">
        <div class="input-group-append" style="cursor: pointer">
            <span
                class="input-group-text bg-transparent border-0 text-white"
                id="savebtn"
                onclick="save_file()"
            >
                <svg
                    xmlns="http://www.w3.org/2000/svg"
                    height="18"
                    width="18"
                    fill="none"
                    viewBox="0 0 24 24"
                    stroke="currentColor"
                    stroke-width="2"
                >
                    <path
                        stroke-linecap="round"
                        stroke-linejoin="round"
                        d="M10 6H6a2 2 0 0-2 2v10a2 2 0 02 2h10a2 2 0 02-2v-4M14
4h6m0 0v6m0-6L10 14"
                    />
                </svg>
                <span> &nbsp;</span>Save
            </span>
        </div>
    </div>
</li>
</ul>

</div>
</div>
</div>
</nav>
<div class="container-fluid">
    <div class="row">
        <div class="col-lg-3 bg-white vh-100 text-black">

```

```

<div class="d-flex justify-content-center py-3">
  <h6>Components</h6>
</div>
<div class="container mb-2 pb-3 border-bottom">
  <div class="row d-flex justify-content-center">
    <button
      name="module"
      class="component btn btn-info col-3 m-1 shadow-md"
    >
      Module
    </button>
    <button
      name="library"
      class="component btn btn-info shadow-lg col-3 m-1"
    >
      Library
    </button>
    <button
      name="control-flow"
      class="component btn btn-info col-3 m-1"
    >
      Control Flow
    </button>
    <button name="data-flow" class="component btn btn-info col-3 m-1">
      Data Flow
    </button>
    <button class="component btn btn-info col-3 m-1">Comp 5</button>
    <button class="component btn btn-info col-3 m-1">Comp 6</button>
  </div>
</div>
<div class="d-flex justify-content-center py-2">
  <h6>Selected Component Attributes</h6>
</div>
<div
  id="selected-comp"
  class="container text-black bg-d3 p-3 shadow-lg formfield"
>
  <!-- pos(x, y), dim(L, W), Label(text), -->
  <label id="sc-id" class="">ID: 100</label> <br />
  <label id="sc-pos" class="">Position: (1000, 1000)</label> <br />
  <label class="">Dimension: </label>
  <input
    id="sc-input-dim"
    type="text"
    style="width: 100%"
    placeholder="Separated by space"
  />
  <br />
  <label class="">Text: </label>
  <textarea id="sc-input-txt" style="width: 100%"></textarea> <br />
  <label id="sc-src-tar">Src & Tar: </label>
  <input

```



```
        style="width: 100%"
        id="sc-input-src-tar"
        placeholder="Id separated by space"
    />
<button id="sc-update" class="btn btn-info mt-2">Update</button>
<button id="sc-delete" class="btn btn-danger mt-2">Delete</button>
<br />
<small
    ><i style="color: rgb(190, 196, 203)"
        >Tip: Right click over component to select it.</i>
    ></small>
    >
</div>
</div>
<div class="col-lg-9 bg-d2 vh-100 d-flex flex-column">
    <div class="d-flex justify-content-center mt-2">
        <h6>Main Editor</h6>
    </div>
    <div class="container m-1 d-flex justify-content-center">
        <div id="root"></div>
    </div>
</div>
</div>
</div>
<footer class="bg-dark text-center text-white">
    <!-- Grid container -->
    <div class="container p-4 pb-0">
        <!-- Section: Social media -->
        <section class="mb-4">
            <!-- Github -->
            <a
                class="btn btn-primary btn-floating m-1"
                style="background-color: #333333;"
                href="https://github.com/deviduttanayak-github/SE-Project-CASE-Tool"
                role="button"
                target="_blank"
            >Github</a>
        </section>
        <!-- Section: Social media -->
    </div>
    <!-- Grid container -->

    <!-- Copyright -->
    <div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2);">
        © 2022 Copyright:
        <a class="text-white" href="#">Team 14 SE Lab</a>
    </div>
    <!-- Copyright -->
</footer>

<!-- dependencies -->
```

```

<script src="fileSaver.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/lodash.js/4.17.21/lodash.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/backbone.js/1.4.0/backbone.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jointjs/3.4.4/joint.js"></script>

<!-- code -->
<script src="component.js"></script>
<script src="main.js"></script>
</body>
</html>

```

### Component.js

```

const TYPE_LINK = "LINK", TYPE_BOX = "BOX_COMP", TYPE_TEXT = "TEXT";

class Component{
  constructor(id, type){
    this.id = id;
    this.type = type;
  }
  display(){
    console.log("Component, id:", this.id, " type:", this.type);
  }
};

class Module extends Component{
  constructor(id, label){
    super(id, TYPE_BOX);
    this.label = label;
    this.pos = [WIDTH/2, HEIGHT/2]
    this.dim = [150, 100]
    this.fillColor = "#e6be53";
    this.color = "#000000";
  }

  display(){
    super.display();
    console.log("Module | label:", this.label)
  }
};

class Library extends Component{
  constructor(id, label){
    super(id, TYPE_BOX);
    this.label = label;
    this.pos = [WIDTH/2, HEIGHT/2]
    this.dim = [180, 100]
    this.fillColor = "#9ae653";

```

```

        this.color = "#000000";
    }
    display(){
        super.display();
        console.log("Library | label:", this.label)
    }
};

class ControlFlow extends Component{
    constructor(id, label){
        super(id, TYPE_LINK);
        this.src = {x : WIDTH/2, y : HEIGHT/2};
        this.tar = {x : this.src.x + 100, y : this.src.y};
        this.color = "#3152e8";
        this.strokeWidth = 3;
        this.label = null;
    }
    display(){
        super.display();
        console.log("Link Control Flow | label: ", this.label, this.src, this.tar);
    }
};

class DataFlow extends Component{
    constructor(id, label){
        super(id, TYPE_LINK);
        this.src = {x : WIDTH/2, y : HEIGHT/2 + 20};
        this.tar = {x : this.src.x + 80, y : this.src.y};
        this.color = "#000000";
        this.strokeWidth = 1;
        this.label = label;
    }
    display(){
        super.display();
        console.log("Link Data Flow | label: ", this.label);
    }
};

// export{
//     Component, Module
// }

```

### Main.js

```

// import { Component, Module } from "./component";

// File handling and other minor things
var file_name = 'Untitled';
var id_tip;
const file_name_container = 'file-name';
const file_name_update = 'change-file-name';
var loadedfjson = false;
const init_setup = () => {

```

```

let nodes = document.getElementsByClassName(file_name_container);
for (let i = 0; i < nodes.length; i++) {
  nodes[i].innerText = file_name;
}
let renamebtn = document.getElementById('rename-file');
// console.log(root, btn);
renamebtn.addEventListener('click', (e) => {
  let name = getfilename();
  if (name !== null) {
    let nodes = document.getElementsByClassName(file_name_container);
    for (let i = 0; i < nodes.length; i++) {
      nodes[i].innerText = name;
    }
  }
});
});

function getfilename() {
  var name = prompt('Please enter file name', 'Untitled');
  if (name !== null && name.trim() !== '') {
    console.log(name);
    file_name = name;
    return name;
  } else {
    alert('Please enter a valid file name');
    return null;
  }
}

// ----- //
// Jointjs window things
var namespace, graph, paper;
const WIDTH = 1000,
      HEIGHT = 600,
      DOM_ID = 'root';

var _ID = 0,
    last_selected = -1;

const get_uid = () => {
  _ID += 1;
  return _ID;
};

var id2comp = new Map();
var id2class = new Map();
var jointId2id = new Map();

const make_rect = (comp) => {
  let posX = comp.pos[0],
      posy = comp.pos[1];
  let L = comp.dim[0],
      W = comp.dim[1];

```

```

let rect = new joint.shapes.standard.Rectangle();
rect.position(posx, posy);
rect.resize(L, W);
rect.attr({
  body: {
    fill: comp.fillColor,
  },
  label: {
    text: comp.label,
    fill: comp.color,
  },
});
return rect;
};

```

```

const make_link = (comp) => {
  console.log('make_link ');
  var link = new joint.shapes.standard.Link();
  link.source(comp.src);
  link.target(comp.tar);
  link.attr({
    line: {
      stroke: comp.color,
      strokeWidth: comp.strokeWidth,
    },
  });
  link.labels([
    {
      attrs: {
        text: {
          text: comp.label,
        },
      },
    },
  ]);
  return link;
};

```

```

const make_component = (comp) => {
  var obj = null;
  if (comp.type == TYPE_BOX) {
    obj = make_rect(comp);
  } else if (comp.type == TYPE_LINK) {
    obj = make_link(comp);
  } else {
    alert('No such Component available yet. ');
  }
  return obj;
};

```

```

const addComponent = (e) => {
  let tt = e.target.name;

```

```

console.log('Somebody clicked me! ', tt);
if (tt == 'module') {
  let id = get_uid();
  comp = make_component(new Module(id, 'Box spawned!'));
  jointId2id[comp.id] = id;
  id2comp[id] = comp;
  id2class[id] = new Module(id, 'Box spawned!');
  id2comp[id].addTo(graph);
} else if (tt == 'library') {
  let id = get_uid();
  let comp = make_component(new Library(id, 'Library spawned!'));
  jointId2id[comp.id] = id;
  id2comp[id] = comp;
  id2class[id] = new Library(id, 'Library spawned!');
  id2comp[id].addTo(graph);
} else if (tt == 'control-flow') {
  let id = get_uid();
  let comp = make_component(new ControlFlow(id, 'None'));
  jointId2id[comp.id] = id;
  id2comp[id] = comp;
  id2class[id] = new ControlFlow(id, 'None');
  id2comp[id].addTo(graph);
} else if (tt == 'data-flow') {
  let id = get_uid();
  let comp = make_component(new DataFlow(id, 'Data'));
  jointId2id[comp.id] = id;
  id2comp[id] = comp;
  id2class[id] = new DataFlow(id, 'Data');
  id2comp[id].addTo(graph);
} else {
  alert('No such component available :(');
}
};

const render_comp_details = (comp) => {
  console.log(comp);
  let cid = comp.id;
  let pos = comp.position(),
  sz =
    comp.attributes.type == 'standard.Rectangle'
      ? comp.size()
      : { width: '--', height: '--' };

  let id = jointId2id[cid];
  console.log(cid, id);
  console.log(comp.attr());
  document.getElementById('sc-id').innerHTML = `ID: ${id}`;
  document.getElementById('sc-pos').innerHTML = `Position: ${pos.x} ${pos.y}`;
  document.getElementById('sc-input-dim').value = `${sz.width} ${sz.height}`;
  if (comp.attributes.type == 'standard.Rectangle')
    document.getElementById('sc-input-txt').value = `${comp.attr().label.text}`;
  else

```

```

document.getElementById('sc-input-txt').value = `${
  comp.labels()[0].attrs.text.text
}`;

if (comp.attributes.type == 'standard.Link') {
  let a = '',
      b = '';
  console.log(comp.source());
  console.log(comp.source().id);
  if (comp.source().id) {
    a = `${jointId2id[comp.source().id]}`;
  } else {
    let p = comp.source();
    a = `(${p.x},${p.y})`;
  }
  if (comp.target().id) {
    b = `${jointId2id[comp.target().id]}`;
  } else {
    let p = comp.target();
    b = `(${p.x},${p.y})`;
  }
  document.getElementById('sc-input-src-tar').value = `${a} ${b}`;
}
last_selected = id;
};

const reset_comp_details = () => {
  document.getElementById('sc-id').innerHTML = `ID: --`;
  document.getElementById('sc-pos').innerHTML = `Position: -- --`;
  document.getElementById('sc-input-dim').value = ``;
  document.getElementById('sc-input-txt').value = ``;
  document.getElementById('sc-input-src-tar').value = ``;
};

const init_window = () => {
  namespace = joint.shapes;
  graph = new joint.dia.Graph({}, { cellNamespace: namespace });
  paper = new joint.dia.Paper({
    el: document.getElementById(DOM_ID),
    model: graph,
    width: WIDTH,
    height: HEIGHT,
    gridSize: 10,
    drawGrid: true,
    cellViewNamespace: namespace,
  });

  paper.on('element:contextmenu', (cellView, e, x, y) => {
    let comp = cellView.model;
    console.log('element:contextmenu : ', cellView, e, x, y);
    render_comp_details(comp);
  });
};

```

```

paper.on('cell:mouseover', (cellView, e, x, y) => {
  let comp = cellView.model;
  console.log('cell:mouseover : ', cellView, e, x, y);
  showID(comp);
});
paper.on('cell:mouseout', (cellView, e, x, y) => {
  console.log('cell:mouseout : ', cellView, e, x, y);
  removeID();
});
paper.on('cell:contextmenu', (cellView, e, x, y) => {
  console.log('cell:contextmenu : ', cellView, e, x, y);
  console.log(cellView.model);
  render_comp_details(cellView.model);
});
paper.on('blank:contextmenu', (cellView, e, x, y) => {
  console.log('blank:contextmenu : ', cellView, e, x, y);
});
paper.on('link:contextmenu', (cellView, e, x, y) => {
  console.log('link:contextmenu : ', cellView, e, x, y);
});

document.getElementById('sc-update').addEventListener('click', (e) => {
  let id = last_selected;
  let comp = id2comp[id];

  if (id > 0) {
    let dim = document.getElementById('sc-input-dim').value;
    let txt = document.getElementById('sc-input-txt').value;

    if (comp.attributes.type == 'standard.Link') {
      let link = document.getElementById('sc-input-src-tar').value;
      console.log('Link ', link);
      try {
        let sid = link.split(' ')[0];
        let tid = link.split(' ')[1];
        if (sid[0] == '(') {
          console.log(sid, tid);
          sid = sid.slice(1, -1);
          tid = tid.slice(1, -1);
          console.log(sid, tid);
          let s = {
            x: parseInt(sid.split(',')[0]),
            y: parseInt(sid.split(',')[1]),
          };
          let e = {
            x: parseInt(tid.split(',')[0]),
            y: parseInt(tid.split(',')[1]),
          };
          console.log(s, e);
          id2comp[id].source(s);
          id2comp[id].target(e);
        } else if (sid > 0 && tid > 0) {

```



```

        id2comp[id].source(id2comp[sid]);
        id2comp[id].target(id2comp[tid]);
    }
    id2comp[id].label(0, {
        attrs: {
            text: {
                text: txt,
            },
        },
    });
} catch (e) {
    console.log(e);
    alert('Bad entry! :(');
}
} else {
    console.log('Rect');
    try {
        let dx = parseInt(dim.split(' ')[0]);
        let dy = parseInt(dim.split(' ')[1]);
        id2comp[id].resize(dx, dy);
    } catch (e) {
        console.log(e);
        alert('Bad entry!!');
    }
    id2comp[id].attr('label/text', txt);
}
}
reset_comp_details();
});
document.getElementById('sc-delete').addEventListener('click', (e) => {
    console.log(last_selected);
    if (last_selected > 0) {
        let id = last_selected;
        id2comp[id].remove();
    }
    reset_comp_details();
});

// binding component to button
let nodes = document.getElementsByClassName('component');
for (let i = 0; i < nodes.length; i += 1) {
    nodes[i].addEventListener('click', (e) => {
        addComponent(e);
    });
}

reset_comp_details();
};

const main = () => {
    console.log('Hello world');
    init_setup();
}

```

```

    init_window();
};

window.addEventListener('load', (e) => {
    main();
});

//easy save and load functions - by utshomax

function loadFile() {
    $('#fileinput').trigger('click');
}
$('#fileinput').on('change', function (e) {
    let file = e.target.files[0].name;
    var input, file, fr;
    if (typeof window.FileReader !== 'function') {
        alert("The file API isn't supported on this browser yet.");
        return;
    }

    input = document.getElementById('fileinput');
    if (!input) {
        alert("Um, couldn't find the fileinput element.");
    } else if (!input.files) {
        alert(
            "This browser doesn't seem to support the `files` property of file inputs."
        );
    } else if (!input.files[0]) {
        alert("Please select a file before clicking 'Load'");
    } else {
        file = input.files[0];
        fr = new FileReader();
        fr.onload = receivedText;
        fr.readAsText(file);
    }

    function receivedText(e) {
        let lines = e.target.result;
        var newArr = JSON.parse(lines);
        graph.fromJSON(newArr);
        let models = graph.attributes.cells.models;
        for(let i=0; i<models.length; i+=1){
            let id = get_uid();
            id2comp[id] = models[i];
            jointId2id[models[i].id] = id;
        }
    }

    let nodes = document.getElementsByClassName(file_name_container);
    for (let i = 0; i < nodes.length; i++) {
        nodes[i].innerHTML = file.split('.')[0];
    }
});

```

```

function save_file() {
  let data = JSON.stringify(graph.toJSON());
  let blob = new Blob([data], { type: 'text/plain;charset=utf-8' });
  saveAs(blob, `${file_name}.json`);
}

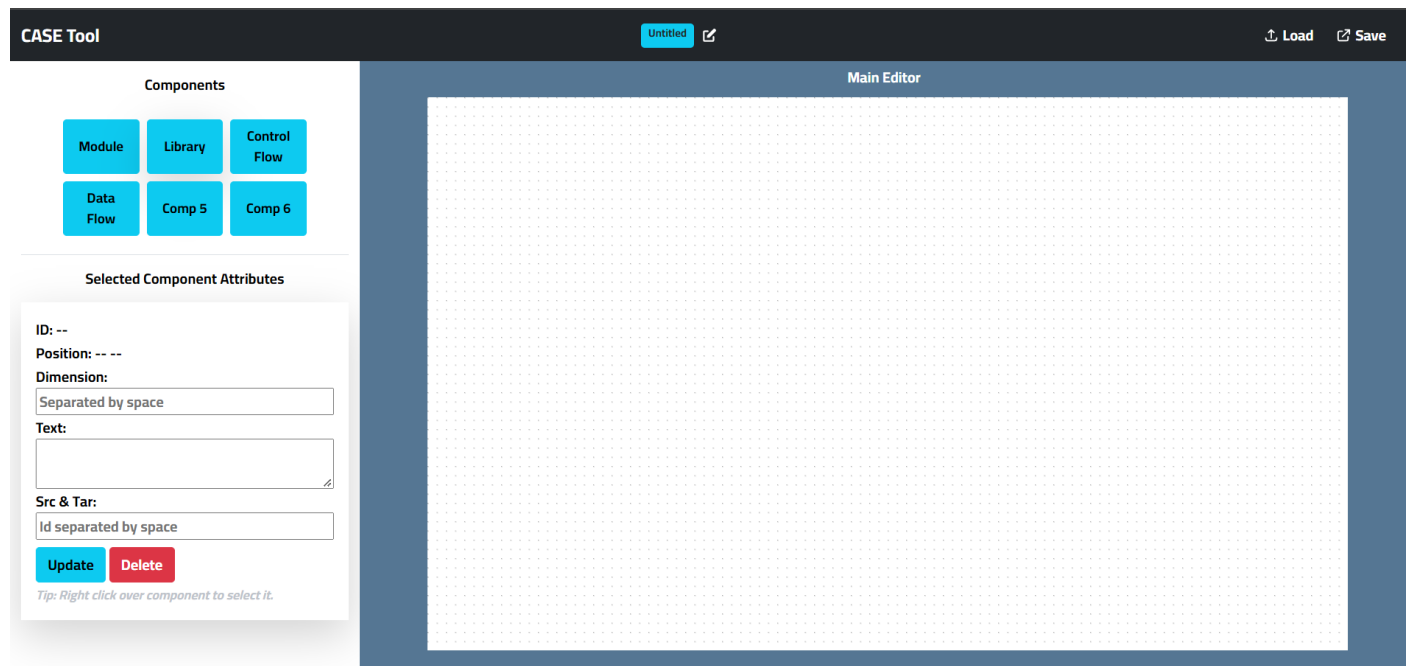
function showID(comp) {
  console.log(comp);
  id_tip = new joint.shapes.basic.Circle({
    position: {
      x: comp.position().x + comp.size().width - 20,
      y: comp.position().y,
    },
    size: { width: 20, height: 20 },
    attrs: { text: { text: jointId2id[comp.id] }, circle: { fill: '#2ECC71' } },
    name: 'id_tip',
  });
  graph.addCell(id_tip);
}

function removeID() {
  if (id_tip) {
    id_tip.remove();
  }
}

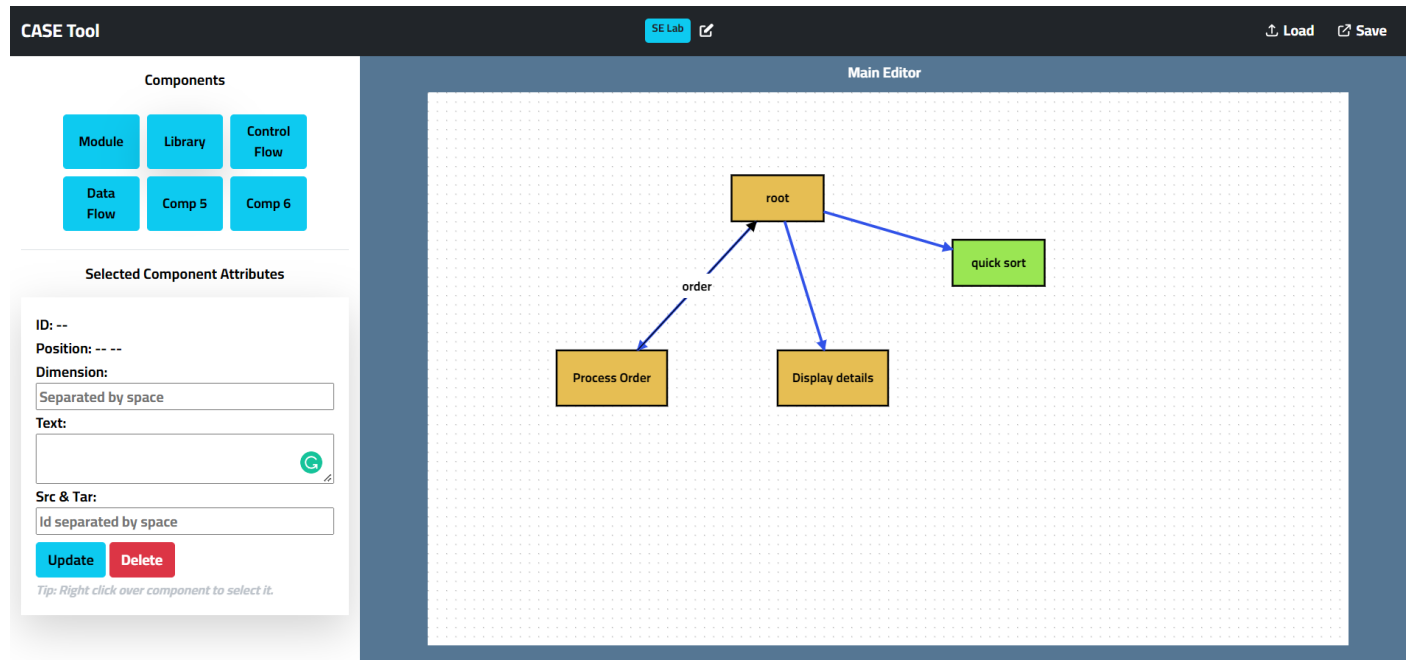
```

# GUI Interface

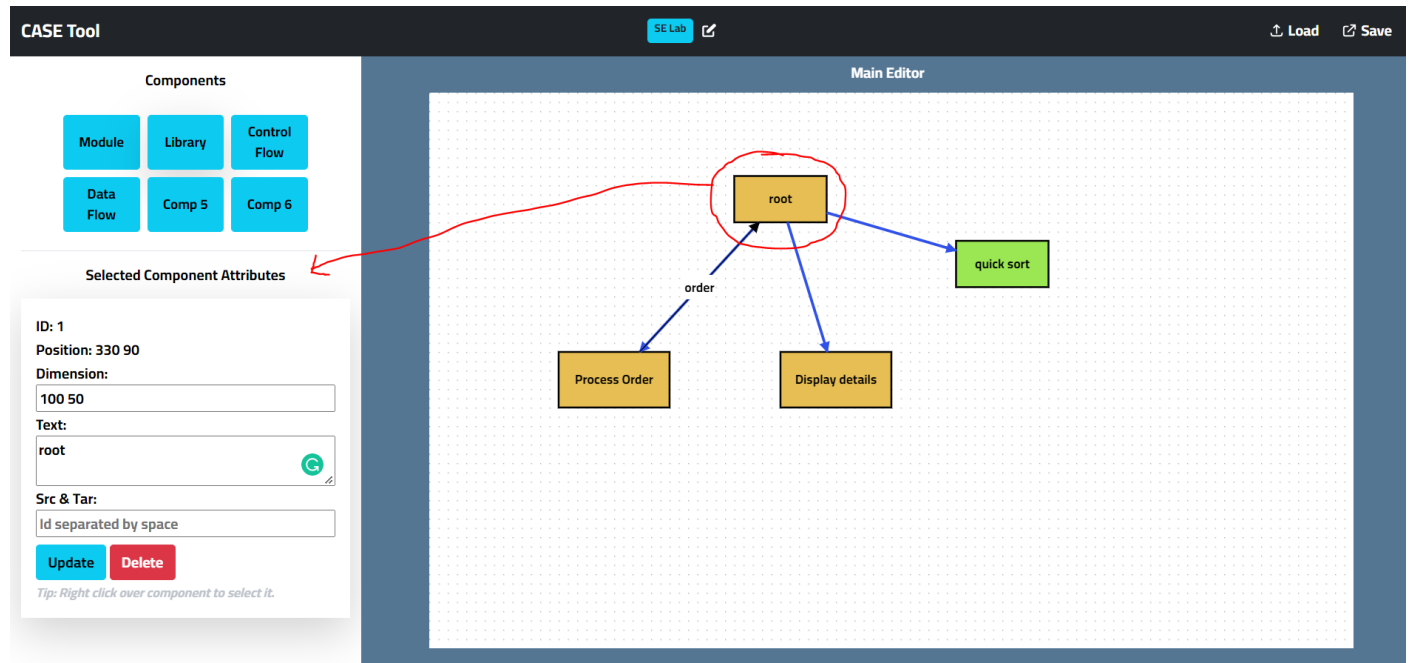
## Main window



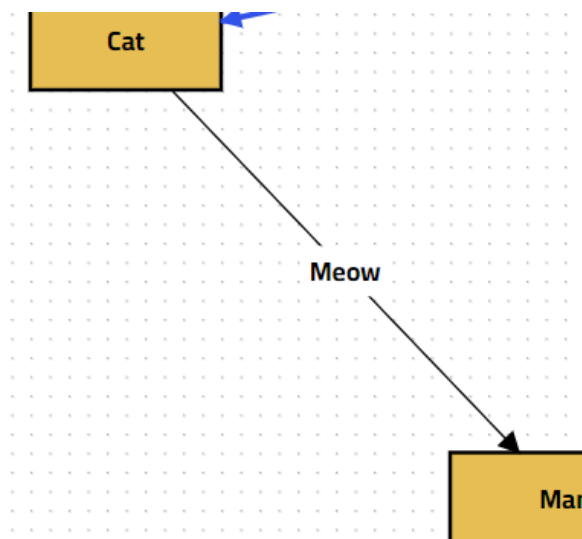
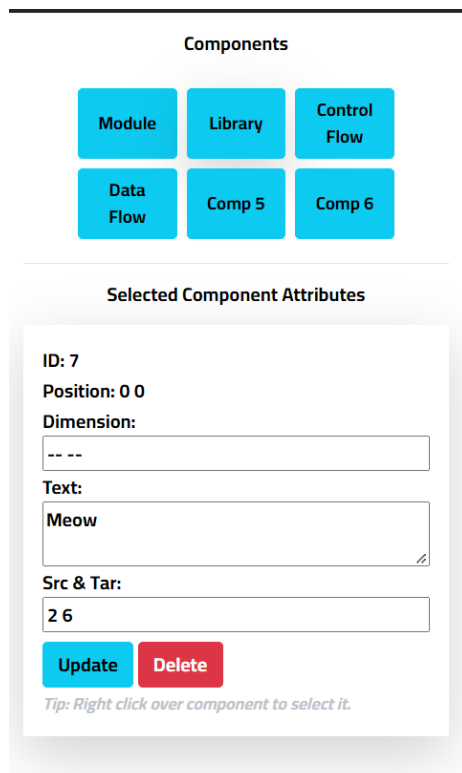
## Main Window with diagram



## Selected Component View



## Arrow Attributes



## **CONTRIBUTION**

### **DEVIDUTTA NAYAK - 119CS0144**

- Source Code
- User Interface
- SRS
- Sequence Diagram

### **UTSAV OJHA (119CS0145)**

- Source Code
- user interface
- SRS
- activity diagram

### **PARESH KUMAR PAGAL (119CS0147)**

- Source Code
- SRS
- user interface
- use case Diagram

### **ADITYA NARAYAN PANDA (119CS0148)**

- Source Code
- SRS
- User Interface
- class diagram