

## **Module 3 Assignment**

Nicholas Kaiser

SE 2300 Software Construction

Dr. Preston

February 19, 2025

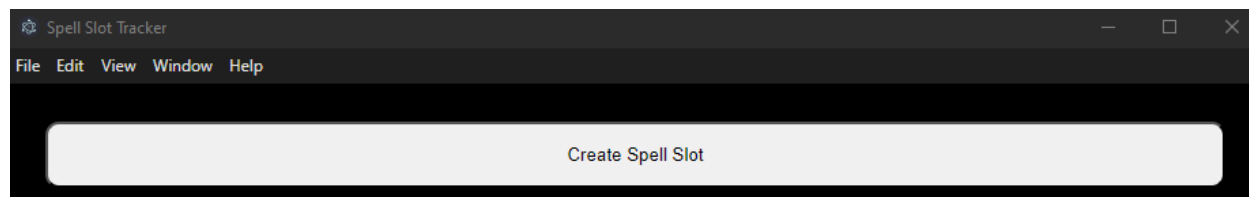
## Functional requirements

Name	Description
<b>F1.0</b>	The system shall have a button to add a spell slot.
<b>F1.1</b>	The system shall have a button inside each slot to select the spell level
<b>F1.2</b>	The system shall allow for slot levels of 0-100
<b>F1.3</b>	The system shall have a button inside each slot to populate the slot with text from a list of spells corresponding with the slot's selected level.
<b>F2.0</b>	The system shall load the list of spells from spell_list.json upon startup
<b>F2.1</b>	The user shall provide the system with a JSON-formatted list
<b>F2.1.1</b>	The spell_list.json shall be formatted as name, level, description
<b>F3.0</b>	The slot shall have a toggle button that shows the slot as spent
<b>F3.1</b>	The toggle button shall undo spent slots
<b>F4.0</b>	The system shall have a button to reset all slots
<b>F5.0</b>	The slots shall display the name, level, and description as provided by the spell_list.json

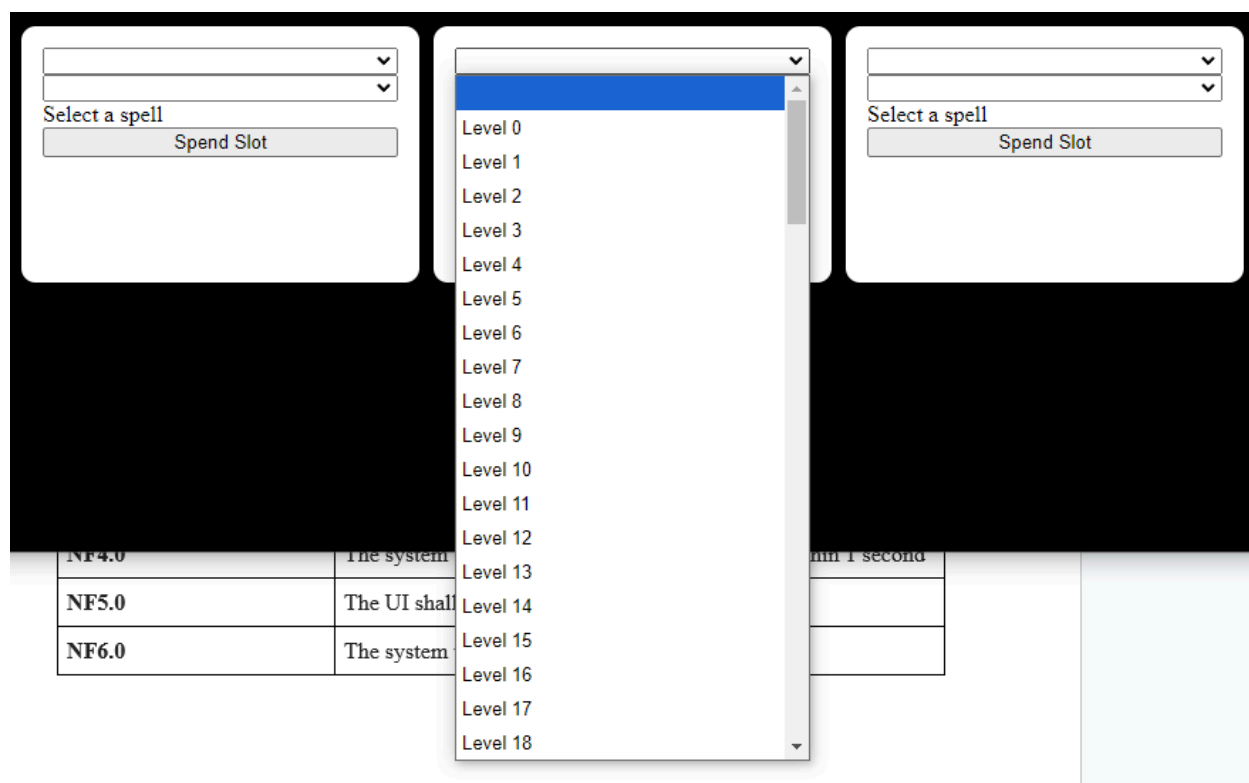
## Non-Functional Requirments

Name	Description
<b>NF1.0</b>	The system shall work with Windows 10 operating systems
<b>NF2.0</b>	The system shall load the spell_list.JSON within 5 seconds
<b>NF3.0</b>	The system shall populate slots within 2 seconds
<b>NF4.0</b>	The system shall toggle spent and unspent slots within 1 second
<b>NF5.0</b>	The UI shall be intuitive
<b>NF6.0</b>	The system text shall use the English language

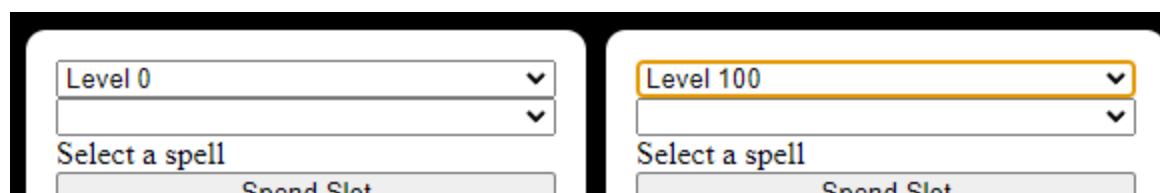
F1.0 The system shall have a button to add a spell slot.



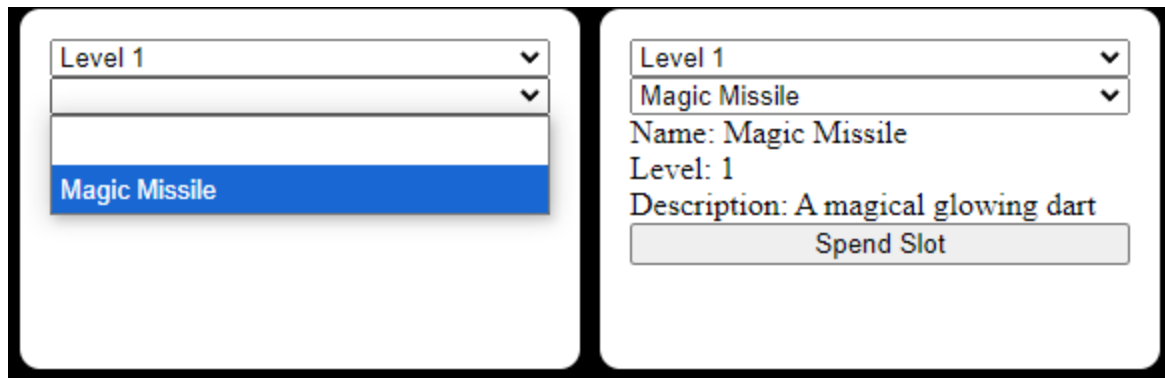
F1.1 The system shall have a button inside each slot to select the spell level



F1.2 The system shall allow for slot levels of 0-100



F1.3 The system shall have a button inside each slot to populate the slot with text from a list of spells corresponding with the slot's selected level.



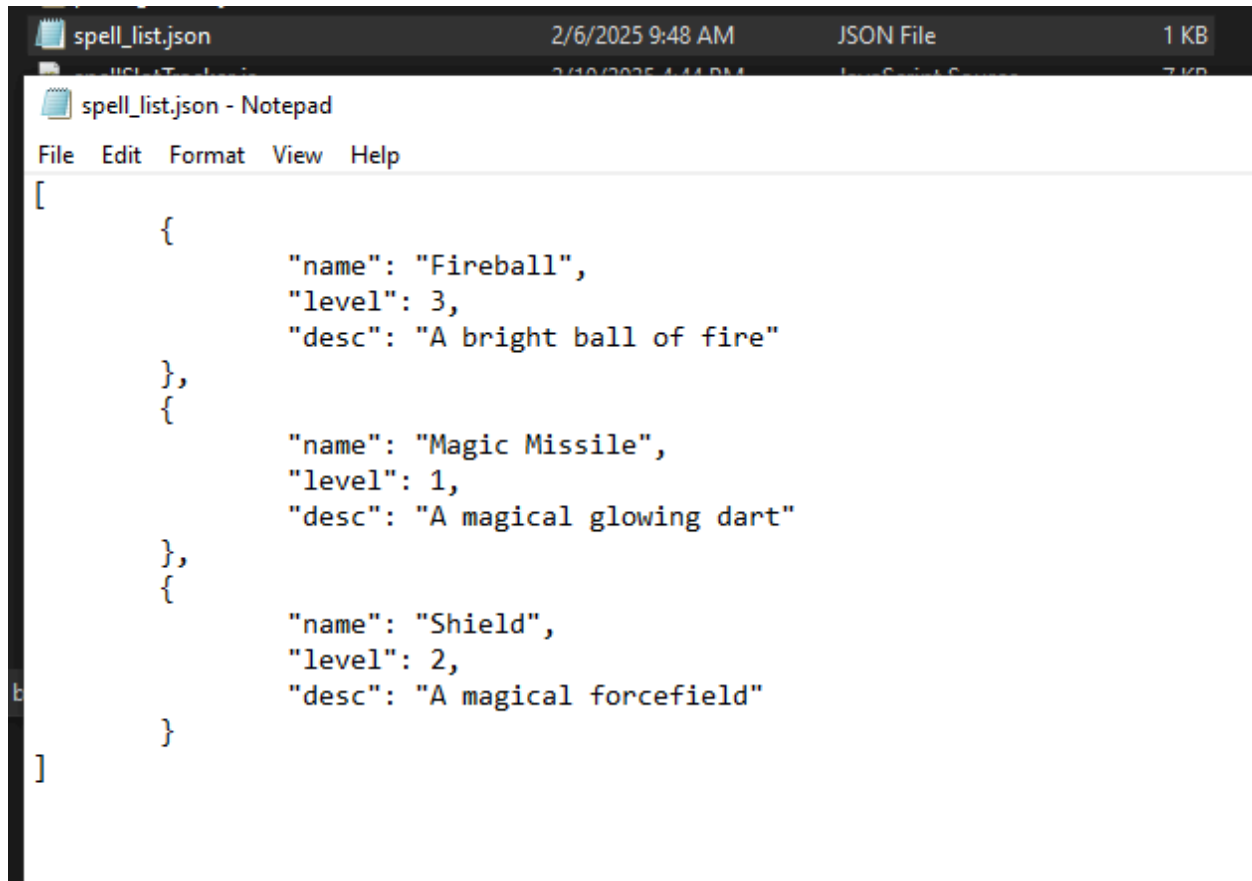
F2.0 The system shall load the list of spells from spell\_list.json upon startup

```
// Create spells and add them to spellList[] for future reference
fetch('spell_list.json').then(response => response.json()).then(data => {
  var count = data.length;
  for (let i = 0; i < count; i++) {
    var spellName = data[i].name; //get name:
    var spellLevel = data[i].level; //get level:
    var spellDesc = data[i].desc; //get description

    var spell = new Spell(spellName, spellLevel, spellDesc);
    spellList.push(spell);
  }
})
```

F2.1 The user shall provide the system with a JSON-formatted list

F2.1.1 The spell\_list.json shall be formatted as name, level, description



```
spell_list.json 2/6/2025 9:48 AM JSON File 1 KB
spell_list.json - Notepad
File Edit Format View Help
[
  {
    "name": "Fireball",
    "level": 3,
    "desc": "A bright ball of fire"
  },
  {
    "name": "Magic Missile",
    "level": 1,
    "desc": "A magical glowing dart"
  },
  {
    "name": "Shield",
    "level": 2,
    "desc": "A magical forcefield"
  }
]
```

F3.0 The slot shall have a toggle button that shows the slot as spent

F3.1 The toggle button shall undo spent slots



F4.0 The system shall have a button to reset all slots



F5.0 The slots shall display the name, level, and description as provided by the spell\_list.json

