

Aetalag Tabletop RPG Asset Manager for UTM CSCI 352

Chase Perritt and Danny Lillard

Abstract

“Aetalag Tabletop RPG Asset Manager” is a modular asset management system for tabletop roleplaying games. The targets for “Aetalag Tabletop RPG Asset Manager” are those who wish to “make it easy” when working with TTRPG assets: dungeon masters and players alike. The project is currently in the development phase, about halfway through now.

1. Introduction

“Aetalag Tabletop RPG Asset Manager” is a modular asset management system for tabletop roleplaying games built to assist the user in TTRPG character building and upkeep via gathering information from online databases and organizing the information in an accessible and dynamic manner. Our target audience is, of course, TTRPG players. We expect the target audience to benefit from “Aetalag Tabletop RPG Asset Manager” due to its ability to better keep track of TTRPG elements in the heat of a game: initiative scores, health points, spell slots, etc.

1.1. Background

Tabletop RPGs, or role-playing games, are an increasingly common hobby in modern society that involves creating and acting out characters in fictional scenarios. One game is typically called a “session,” and multiple sessions are called a “campaign.” One such example of a tabletop role-playing game is the widely popular Dungeons and Dragons (5th Edition). D&D [5e] sessions tend to take place in high fantasy, magical settings, however, the rules governing them are quite real and concrete. Some terms that would be useful to know are as follows:

Sheet: A page in a character’s portfolio, typically used to track their stats, equipment, and/or spells.

Initiative: This is an integer that is used to determine when an entity is allowed to do actions in the list of all other entities. For example, a ghost with initiative 3 goes after a person with initiative 4.

Spell Slots: If a spell slot is a chamber of a revolver, the spell is the round. Each player has a certain number of spell slots, based on various factors. These spell slots are leveled, corresponding to the level of the spell. So a user can have five level one spells, but only one level one slot. And thus can only do one level one spell per long rest. Spell slots are reset by a long rest.

Dice: D&D is ruled by seven dice, in order of sides: 4, 6, 8, 10, 12, and 20 sided dice. The player is expected to use these dice in real life. Within the program, the syntax to represent dice is: d-sideNum.

1.2. Challenges

A particular challenge in mapping out both the ruleset of D&D 5e is that, though we have access to source reference documents and particular websites that compile all of the information from several source books and pdfs, there is simply too much information out there. This is a gargantuan undertaking, and a simple solution would simply be to add less content than we hope: to cut out the equipment databases and alternate features in favor of the minimum required amount.

2. Scope

Completion of the project entails the existence of a D&D 5e character maintenance tool that rivals traditional pen and paper sheets in terms of quality of life assurance and ease of access to information. While this software may not render the pen and paper style obsolete, it should prove valuable as a beginner’s introductory tool to tabletop roleplaying or as a reference for rules and monster stats for Dungeon Masters. As for stretch goals we would like to give the user to save their sheet at a .pdf or .txt, and have an automatic initiative roller.

2.1. Requirements

Most requirements for “Aetalag Tabletop RPG Asset Manager” are derived directly from the standard pen and paper style of managing a character. Each requirement listed below deserves to be here because it makes the player’s life easier, whether that requirement does something better than pen or paper, or adds new functionality.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Closing the Program	User	Low	1
2	Adding a New Entity to "HP Tracker"	User	Low	1
3	Deleting an entity from "HP Tracker"	User	Low	1
4	Editing an entity in "HP Tracker"	User	Low	1
5	Adding a new entity to "Initiative Tracker"	User	Low	1
6	Deleting an entity from "Initiative Tracker"	User	Low	1
7	Editing an entity in "Initiative Tracker"	User	Low	1
8	Sorting entities in "Initiative Tracker"	User	Low	1
9	Creating a Blank Character Sheet	User	Low	1
10	Opening an Old Sheet	User	Low	1
11	Creating a Sheet With "Make it Easy"	User	Low	1
12	Saving a Sheet	User	Low	1
13	Saving a Sheet As	User	Low	1
14	Editing Sheet Values	User	Low	1
15	Changing Drop Down Choices	User	Low	1

TABLE 1. SAMPLE USE CASE TABLE

2.1.1. Functional.

- Users are able to create new characters - By clicking on the "New Sheet" button on the "Aetalag Tabletop RPG Asset Manager" window the user is able to create a new character from scratch
- Users are able to open old sheets - By clicking the "Open Sheet" button on the "Aetalag Tabletop RPG Asset Manager" window the user is able to open a saved sheet
- User is able to use the "Make it Easy" functionality to create a new character - By selecting the "Make it Easy" button on the "Aetalag Tabletop RPG Asset Manager" the user is able to quickly make a character
- Users are able to keep track of the HP of entities - Using the "HP Tracker" subsection of the "Aetalag Tabletop RPG Asset Manager" users should be able to add new entities, edit their values and name, and remove entities from the subsection
- Users are able to keep track of the initiative scores of entities - Using the "Initiative Tracker" subsection of the "Aetalag Tabletop RPG Asset Manager" users are able to track the initiative of entities. They should be able to add entities, move the initiative along, name the entities, and delete them as well
- Users are able to request more information on the HP, AC, Speed, Saves, and Initiative blocks from the "Character Sheet" window - By clicking on the dog ears present on these information block users are able to request more information about the blocks and edit information inside
- Users are able to take long and short rests - By clicking the "Short Rest" and or "Long Rest" buttons present on the "Character Sheet" window users can take rests, being prompted by a window
- Users need to have a save file for each character - analogous to a sheet of paper this will keep track of all his or her last saved values for said character
- Users need to be able to change the values of a character - through text boxes, he or she will be able to alter the character's values, after unlocking said boxes
- The software must not allow incorrect input - users should not be able to set their health too high, raise or lower skill levels to strange numbers, etc.
- The software must have all the important information found in the user's handbook - each part of the character should be laid out and described for the player, if so desired
- Different tabs for different information - a character is usually complex, having backstory, ability stores/feats, class and level, spells, proficiencies and an inventory. This data should be broken down logically into the different tabs: main, skills, equip, spells, ability
- Users have the ability to take notes - note taking is an important part of TTRPGs there should be a separate tab for these notes in the program

2.1.2. Non-Functional.

- Interchangeability - The program is easy for the developers to adapt to new TTRPG frameworks
- Speed of loading and saving - previous saves and the act of saving must load quickly for the user, in less than two seconds

2.2. Use Cases

Use Case Number: 1

Use Case Name: Closing the Program

Description: The use wants to close out of the program.

- User selects the “x” button on “Aetalag Tabletop RPG Asset Manager” window.
- User is prompted by a window, the “Close Program” window. The “Close Program” window has the message:
”Are you sure you want to leave?”
With these three buttons: ”Save and Exit” ”Exit Without Saving” and ”Cancel”

Termination Outcome: If the user selects “Save and Exit”, all open sheets are saved, then the program closes all open windows. If the user selects “Exit Without Saving”, the program closes all open windows. If the user selects “Cancel” the “Close Program” window is closed.

Alternative: User selects the “x” button on the “Close Program” window.

- User selects the “x” button on the “Close Program” window.

Termination Outcome: The “Close Program” window is closed, similar to the “Cancel” button.

Use Case Number: 2

Use Case Name: Adding a new entity to “HP Tracker”

Description: User wants to create a new entity to the “HP Tracker” section on the “Aetalag Tabletop RPG Asset Manager” window. Allowing them to quickly view HP.

- User Selects the “+” button present to the top right of the “HP Tracker” section of the “Aetalag Tabletop RPG Asset Manager” window.
- A new HP bar appears with the fillable fields: “Enter Name” and “0/0” as current and max health, which each can be edited.
- User fills out these fields.

Termination Outcome: User has a new filled HP tracker.

Alternative: User Adds a Sheet entity to “HP Tracker”

- User Selects “Add to HP Tracker” on the sheet.
- The entity is added to “HP Tracker”.

Use Case Number: 3

Use Case Name: Deleting an entity from “HP Tracker”

Description: User wants to delete an entity from the “HP Tracker” section on the “Aetalag Tabletop RPG Asset Manager” window.

- User selects the “X” button present in the block with that “HP Tracker” object.
- That item is removed from the list.

Termination Outcome: User has one less item in the “HP Tracker”.

Use Case Number: 4

Use Case Name: Editing an entity in “HP Tracker”

Description: User wants to edit an entity from the “HP Tracker” section on the “Aetalag Tabletop RPG Asset Manager” window.

- User selects the the item they want to select, currentHP, MaxHP, or Name, present in the block with that “HP Tracker” object.
- The user types in the new value.
- The value is changed.

Termination Outcome: User has a new value.

Use Case Number: 5

Use Case Name: Adding a new entity to “Initiative Tracker”

Description: User wants to create a new entity to the “Initiative Tracker” section on the “Aetalag Tabletop RPG Asset Manager” window. Allowing them to track that entities initiative

- User selects the “+” button present to the top right of the “Initiative Tracker” section of the “Aetalag Tabletop RPG Asset Manager” window.
- A new initiative tracker appears with the fillable fields: “Enter Name” and “0” as the initiative, which each can be edited.
- User fills out these fields.

Termination Outcome: User has a new filled initiative tracker.

Alternative: User Adds a Sheet entity to “Initiative Tracker”

- User Selects “Add to Initiative Tracker” on the sheet.
- The entity is added to “Initiative Tracker”.

Use Case Number: 6

Use Case Name: Deleting an entity from “Initiative Tracker”

Description: User wants to delete an entity from the “Initiative Tracker” section on the “Aetalag Tabletop RPG Asset Manager” window.

- User Selects the “X” button present in the block with that “Initiative Tracker” object.
- That item is removed from the list.

Termination Outcome: User has one less item in the “Initiative Tracker”.

Use Case Number: 7

Use Case Name: Editing an entity in “Initiative Tracker”

Description: User wants to edit an entity from the “Initiative Tracker” section on the “Aetalag Tabletop RPG Asset Manager” window.

- User Selects the item they want to select, Initiative or Name, present in the block with that “Initiative Tracker” object.
- The user types in the new value.
- The value is changed.

Termination Outcome: User has a new value.

Use Case Number: 8

Use Case Name: Sorting entities in “Initiative Tracker”

Description: User wants to sort entities in the “Initiative Tracker” section on the “Aetalag Tabletop RPG Asset Manager” window.

- User fills in the Initiative of each item with a value.
- The user clicks the “Sort” button present on the “Initiative Tracker”
- The values are sorted, low to high.

Termination Outcome: User has a sorted list.

Use Case Number: 9

Use Case Name: Creating a Blank Character Sheet

Description: The user wants to create a new character from scratch (without using the “Make It Easy” functionality).

- User Selects “New Sheet” button on the “Aetalag Tabletop RPG Asset Manager” window.
- A blank sheet is opened in a new window.
- The user will then be able to fill out any blank fields in the sheet.

Termination Outcome: User has a new filled sheet.

Alternative: Sheet Manager window is already open

- User Selects “New” on the “Aetalag Tabletop RPG Asset Manager” window.
- A blank sheet is opened in a new window.
- The user will then be able to fill out any blank fields in the sheet.

Use Case Number: 10

Use Case Name: Opening an Old Sheet

Description: The user wants to open a sheet they have been already working on.

- User Selects “Open Sheet” button on the “Aetalag Tabletop RPG Asset Manager” window.
- A windows file explorer screen is opened.
- The user navigates to the .aetalag file they wish to open.

Termination Outcome: The sheet is loaded.

Use Case Number: 11

Use Case Name: Creating a Sheet With “Make it Easy”

Description: The user wants to create a new sheet, the easy way.

- User Selects “Make it Easy” button on the “Aetalag Tabletop RPG Asset Manager” window.
- The “Make it Easy” window is opened.
- The user selects the drop down menu items, and fills in the values that they want.

Termination Outcome: A sheet is created with the users selected values.

Use Case Number: 12

Use Case Name: Saving a Sheet

Description: The user wants to save their sheet.

- User Selects “Save” button on “Sheet” window.
- The sheet values are complied and saved.

Termination Outcome: The .aetalag file is updated.

Alternative: The sheet is not saved at all.

- The “Save As” protolog runs.

Use Case Number: 13

Use Case Name: Saving a Sheet As

Description: The user wants to save a sheet under a new name.

- User Selects “Save As” button on “Sheet” window.
- A windows file explorer window is opened.
- User saves their file as a .aetalag file.
- The sheet values are compiled and saved.

Termination Outcome: The .aetalag file is created.

Most of the following use cases will be general cases. I do not believe that breaking down each part any further will be beneficial to anyone.

Use Case Number: 14

Use Case Name: Editing Sheet Values

Description: The user wants to edit values in their sheet. This use case is a general catch for all the different things that can be edited, there are far too many to have a case for each. The rule is that if the data is encased in a text box it can be edited.

- User Selects a TextBox in their “Sheet”
- The user types in a new value.

Termination Outcome: The value is changed.

Use Case Number: 15

Use Case Name: Changing Drop Down Choices

Description: The user wants to change a value in a drop down menu.

- User Selects the drop down.
- The user selects a value in the drop down.

Termination Outcome: The value is changed.

2.3. Interface Mockups

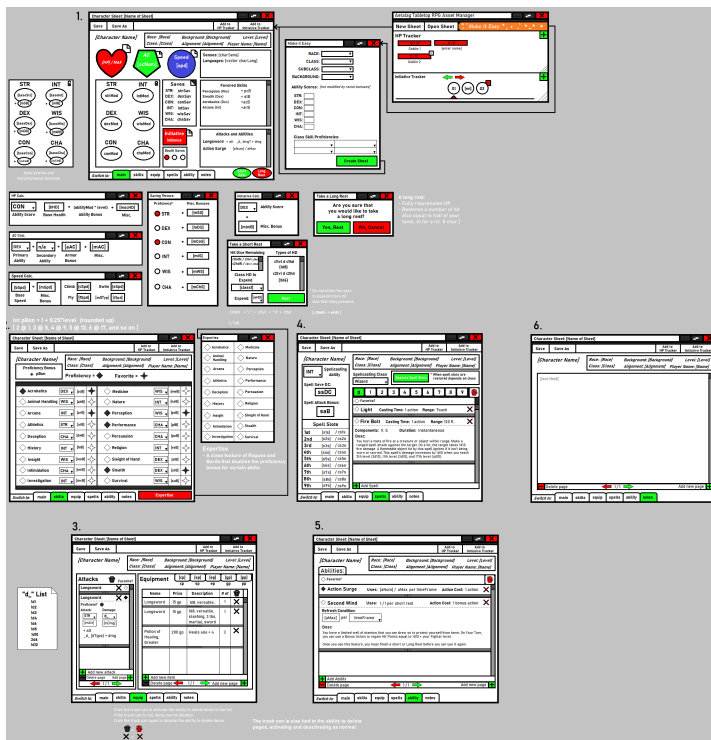


Figure 1. This is the entire user interface laid out.

3. Project Timeline

1) Requirements:

We plan to have all of our requirements established by the submission and review of the first project

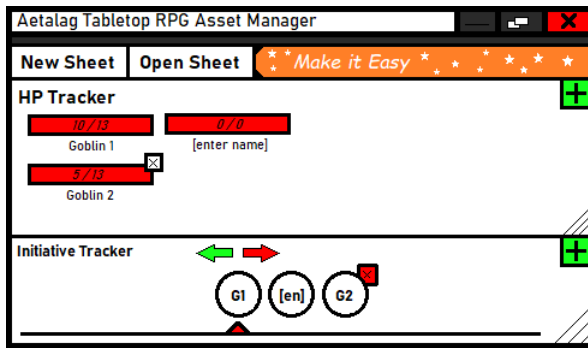


Figure 2. This is the window that the user will be welcomed with. Allowing them to take paths to all functionality. Has use cases 1, 2, and 3.

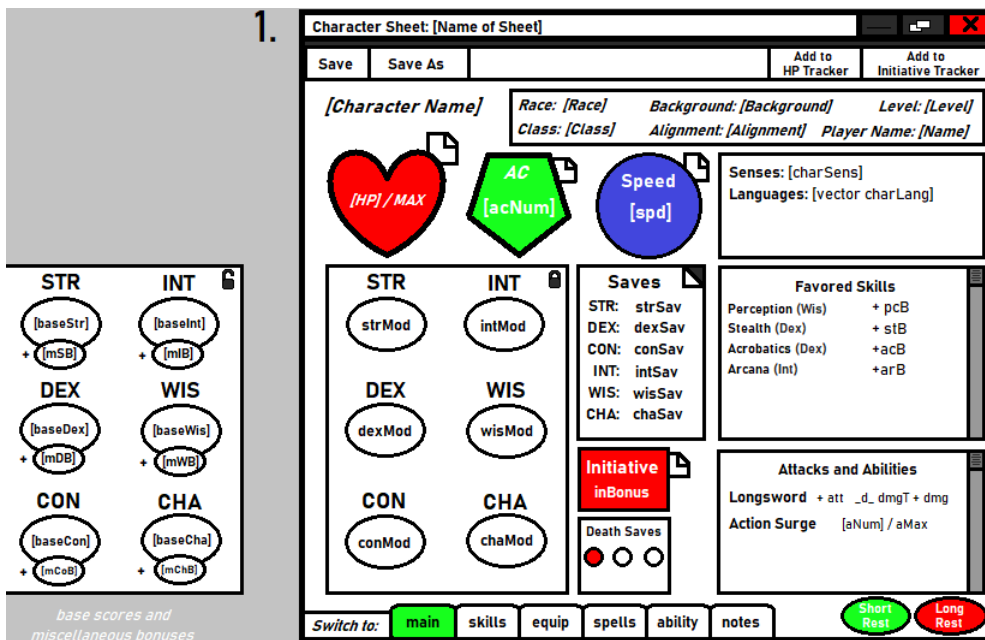


Figure 3. This is the window that the user will use to interact with the most of their characters information. This is the result of use case 3.

proposal draft on the 10th of February, 2020.

2) Design:

We plan for our major design decisions to be fleshed out by March 12th.

3) Implementation:

We hope to have finished implementing the primary, promised functionality of “Aetalag Tabletop RPG Asset Manager” by April 21st.

4) Verification:

We hope to have fixed all bugs and leaks by April 21st.

5) Maintenance:

After the end of April, any additional functionality and systems could be added at any point in the future.

4. Project Structure

Beginning the program the user is greeted by the Asset Manager window. This window is used to: track HP, track initiative, open sheets, and create new sheets, using “New Sheet” or “Make It Easy.”

Once the user has opened a sheet the project is broken down Sheet → Tabs. Each tab calls on the sheet. If the tab needs to know ability score is will go through the Strategy/Observer pattern created in this project.

Make it Easy

RACE:

CLASS:

SUBCLASS:

BACKGROUND:

Ability Scores: *[not modified by racial bonuses]*

STR:

DEX:

CON:

INT:

WIS:

CHA:

Class Skill Proficiencies:

Create Sheet

Figure 4. This is the window that the user will use to create a character easily.

HP Calc.

CON + [bHD] + (abilityMod * level) + [mscHD]

Ability Score Base Health Ability Bonus Misc.

AC Calc.

DEX + n/a + [aAC] + [mAC]

Primary Ability Secondary Ability Armor Bonus Misc.

Speed Calc.

[bSpd] + [mSpd] Cimb [cSpd] Swim [sSpd]

Base Speed Misc. Bonus Fly [fSpd] [mSTyp] [lSpd]

Figure 5. These are the windows that are present from the dog eared, HP, AC, and Speed buttons.

Saving Throws

Proficiency? Misc. Bonuses

☒ STR + [mSS]

☐ DEX + [mDS]

☒ CON + [mCoS]

☐ INT + [mIS]

☐ WIS + [mWS]

☐ CHA + [mChS]

Figure 6. This window comes from the dog ear of the “Saving Throws” subsection of the Character Sheet.

The Skills, Equip, Spells, and Notes Tabs all make use of some variation of the Tab → Window → Object structure. This structure is used to standardize the creation and use of many object in the program.

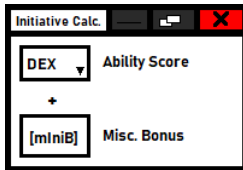


Figure 7. This window comes from the dog ear of the “Initiative” subsection of the Character Sheet.

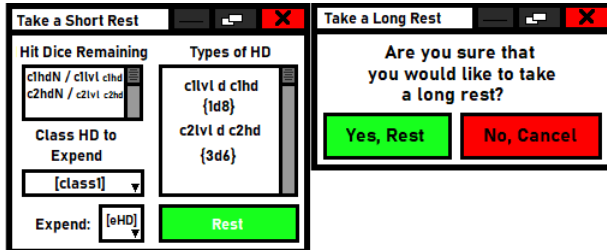


Figure 8. These windows come from the “Short Rest” and “Long Rest” buttons on the “Character Sheet” main window

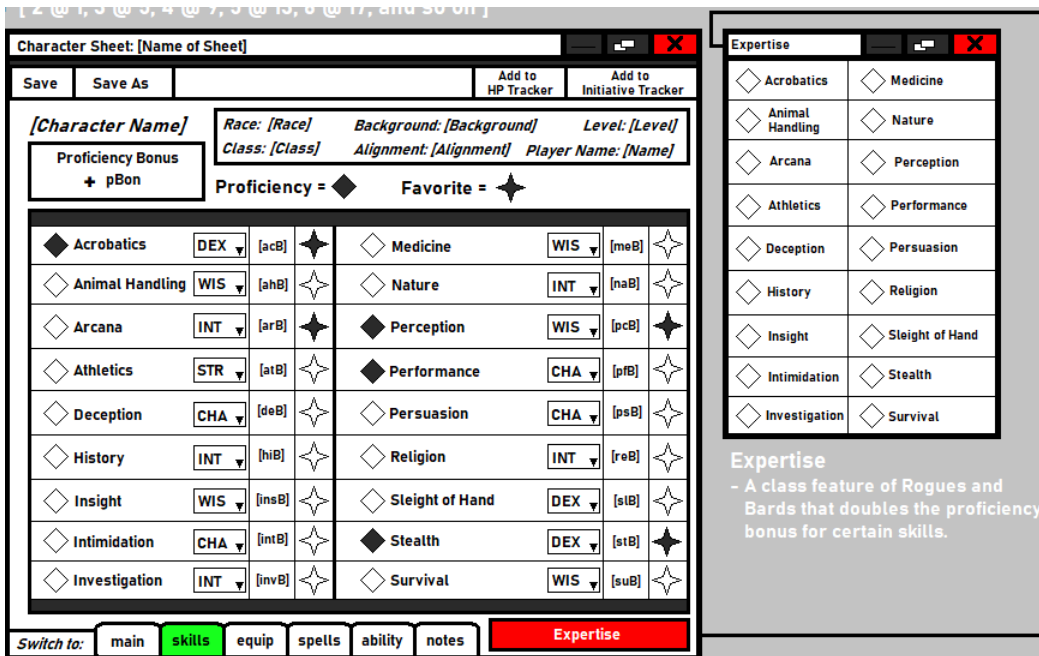


Figure 9. This window comes from the “Skills” tab of the “Character Sheet” window.

4.1. UML Outlines

4.2. Design Patterns Used

Firstly, the MVVM design pattern is used in order to dynamically update the HP and Initiative Tracker on the “Atealag Tabletop RPG Asset Manager” window. This design pattern is a perfect solution to our problem of needing to dynamically update a list of elements. Secondly, we use a publisher-subscriber design pattern to dynamically update stats that work off of ability scores. Making it such that any change to an ability score will automatically update all stats dependent on it. And also allows said stats to quickly decouple from an ability score and couple to another.

Character Sheet: [Name of Sheet]

Save

Save As

Add to HP Tracker

Add to Initiative Tracker

[Character Name]

Race: [Race]

Background: [Background]

Level: [Level]

Class: [Class]

Alignment: [Alignment]

Player Name: [Name]

Attacks

Longsword

Longsword

Proficient?

Attack:

Damage:

STR

d_

[mAtt]

[mDmg]

+ att

d [dType] + dmg

Add new attack

Delete page

Add page

1 / 1

Equipment

[cp]

[sp]

[ep]

[gp]

[pp]

Name	Price	Description	# of	
Longsword	15 gp	1d8, versatile..	1	X
Longsword	15 gp	1d8, versatile, slashing, 3 lbs, martial, sword	1	X
Potion of Healing, Greater	200 gp	Heals 4d4 + 4	3	X

Add new item

Delete page

1 / 1

Add new page

Switch to:

main

skills

equip

spells

ability

notes

"d_" List

1d1

1d2

1d3

1d4

1d6

1d8

1d10

2d6

1d12

Figure 10. This window comes from the "Equip" tab of the "Character Sheet" window.

Character Sheet: [Name of Sheet]

Save

Save As

Add to HP Tracker

Add to Initiative Tracker

[Character Name]

Race: [Race]

Background: [Background]

Level: [Level]

Class: [Class]

Alignment: [Alignment]

Player Name: [Name]

INT

Spellcasting Ability

Spell Save DC:

ssDC

Spell Attack Bonus:

saB

Spell Slots

1st	[s1s]	/ cs1s
2nd	[s2s]	/ cs2s
3rd	[s3s]	/ cs3s
4th	[s4s]	/ cs4s
5th	[s5s]	/ cs5s
6th	[s6s]	/ cs6s
7th	[s7s]	/ cs7s
8th	[s8s]	/ cs8s
9th	[s9s]	/ cs9s

Spellcasting Class

Wizard

Restore Spell Slots

* When spell slots are restored depends on class

0

1

2

3

4

5

6

7

8

9

Favorite?

Light

Casting Time: 1 action

Range: Touch

Fire Bolt

Casting Time: 1 action

Range: 120 ft.

Components: V, S

Duration: Instantaneous

Desc:

You hurl a mote of fire at a creature or object within range. Make a ranged spell attack against the target. On a hit, the target takes 1d10 fire damage. A flammable object hit by this spell ignites if it isn't being worn or carried. This spell's damage increases by 1d10 when you reach 5th level (2d10), 11th level (3d10), and 17th level (4d10).

Add Spell

Switch to:

main

skills

equip

spells

ability

notes

Figure 11. This window comes from the "Spells" tab of the "Character Sheet" window.

5. Results

As of handin four Chase and I have designed the project, both in UML and in GUI. Now all is needed is to begin development. As of handin five Chase and I have begun development. Right now we are about halfway through.

5.1. Future Work

Right now it is time for frantic, and fast-paced development.

Character Sheet: [Name of Sheet]

Save
Save As

Add to HP Tracker
Add to Initiative Tracker

[Character Name]

Race: [Race]
Background: [Background]
Level: [Level]
Class: [Class]
Alignment: [Alignment]
Player Name: [Name]

Abilities:

Favorite?

Action Surge
Uses: [aNum] / aMax per timeFrame
Action Cost: 1 action

Second Wind
Uses: 1 / 1 per short rest
Action Cost: 1 bonus action

Refresh Condition:

[aMax]
per
timeFrame

Desc:

You have a limited well of stamina that you can draw on to protect yourself from harm. On Your Turn, you can use a Bonus Action to regain Hit Points equal to 1d10 + your Fighter level.

Once you use this feature, you must finish a short or Long Rest before you can use it again.

Add Ability
Delete page

1 / 1

Add new page

Switch to:
main
skills
equip
spells
ability
notes

Figure 12. This window comes from the “Ability” tab of the “Character Sheet” window.

Character Sheet: [Name of Sheet]

Save
Save As

Add to HP Tracker
Add to Initiative Tracker

[Character Name]

Race: [Race]
Background: [Background]
Level: [Level]
Class: [Class]
Alignment: [Alignment]
Player Name: [Name]

[text field]

Delete page

1 / 1

Add new page

Switch to:
main
skills
equip
spells
ability
notes

Figure 13. This window comes from the “Notes” tab of the “Character Sheet” window.

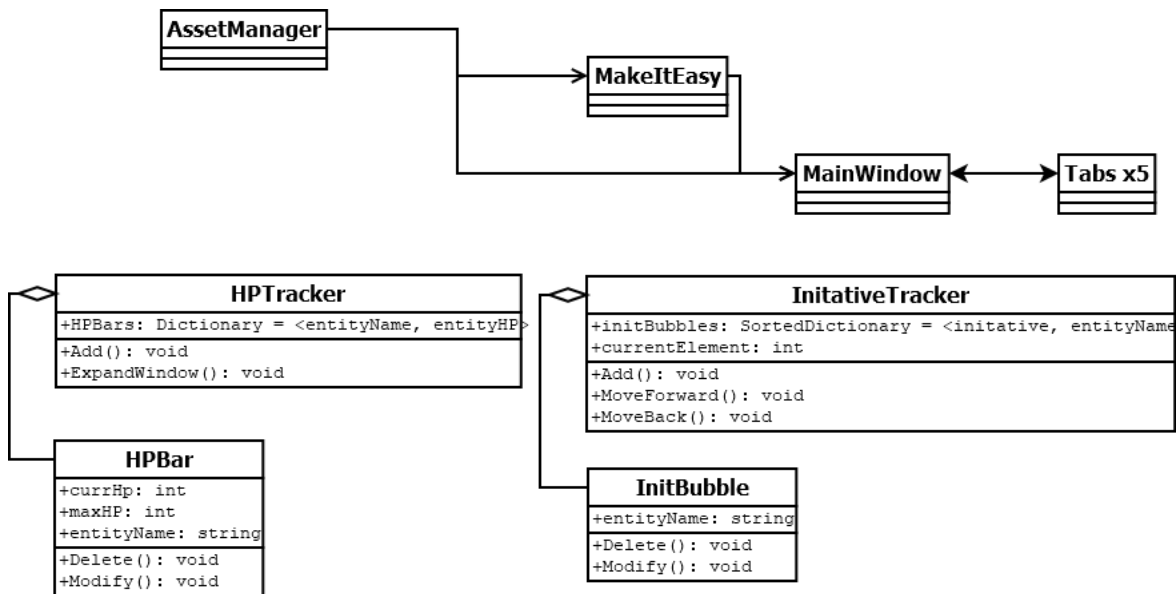


Figure 14. UML for the Asset Manager window.

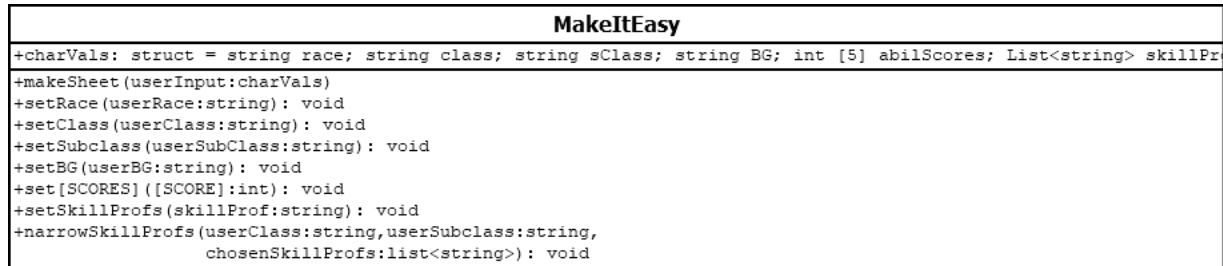


Figure 15. UML for the Make It Easy window.

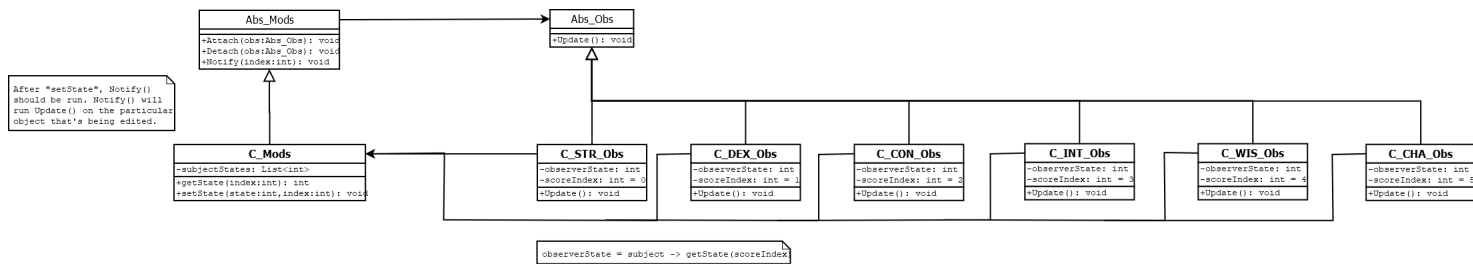


Figure 16. UML for a mesh between the Strategy and Observer Pattern.

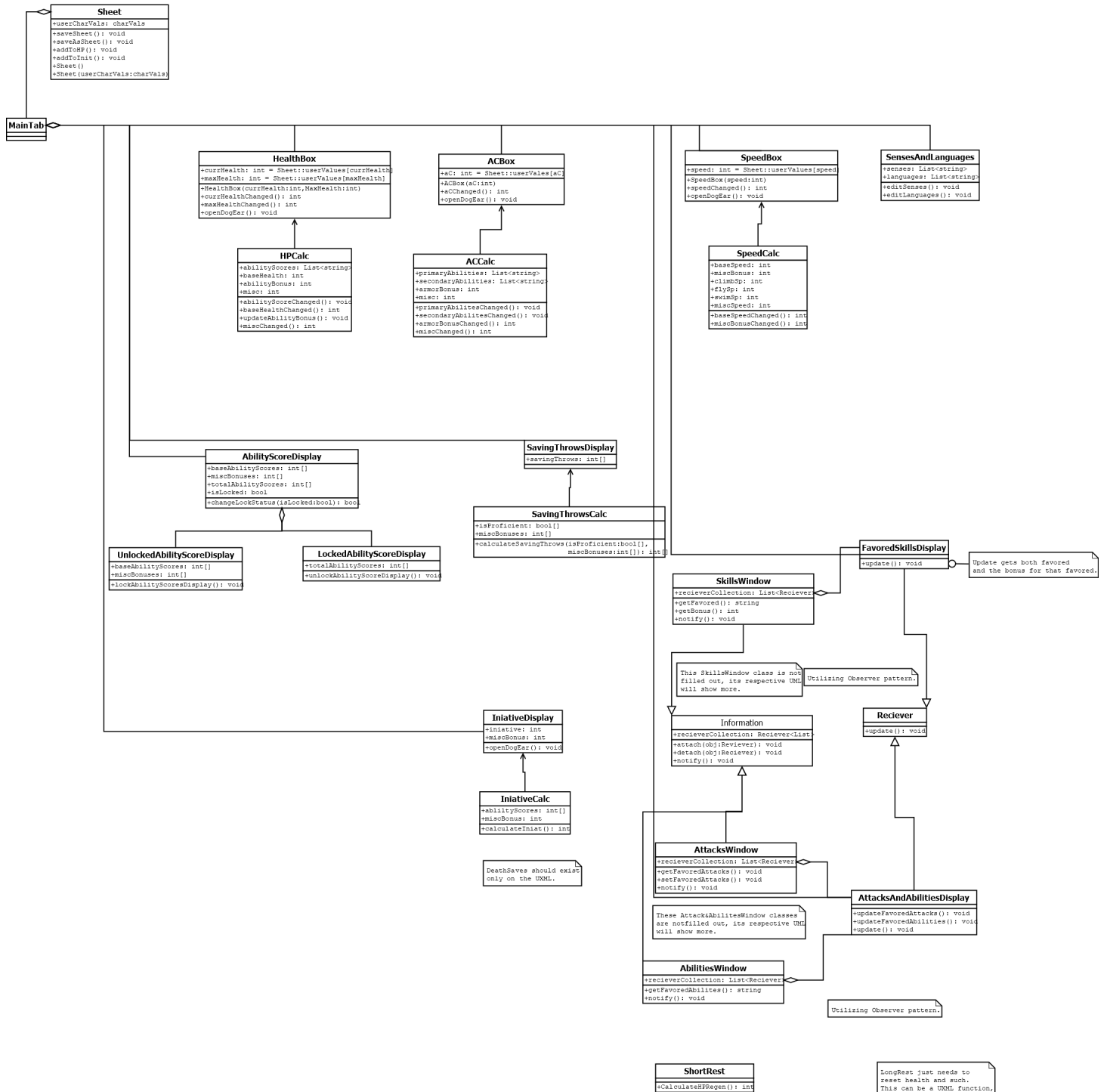


Figure 17. UML for the Main Tab window.

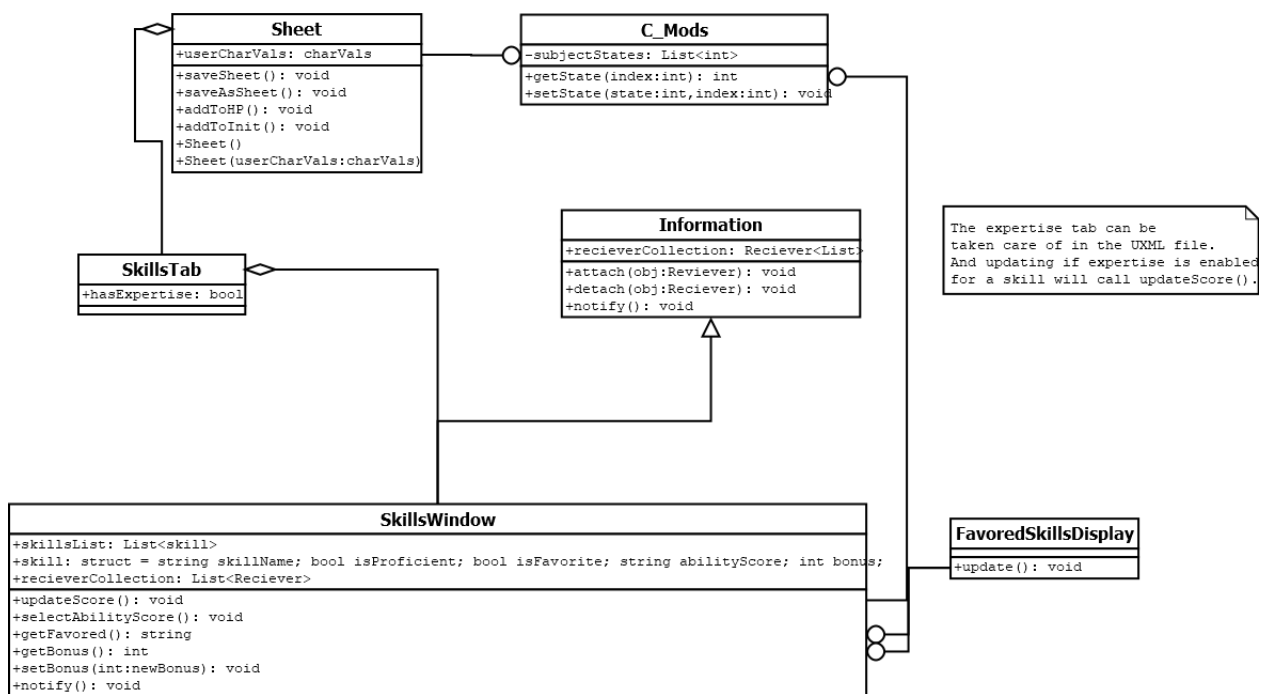


Figure 18. UML for the Skills Tab window.

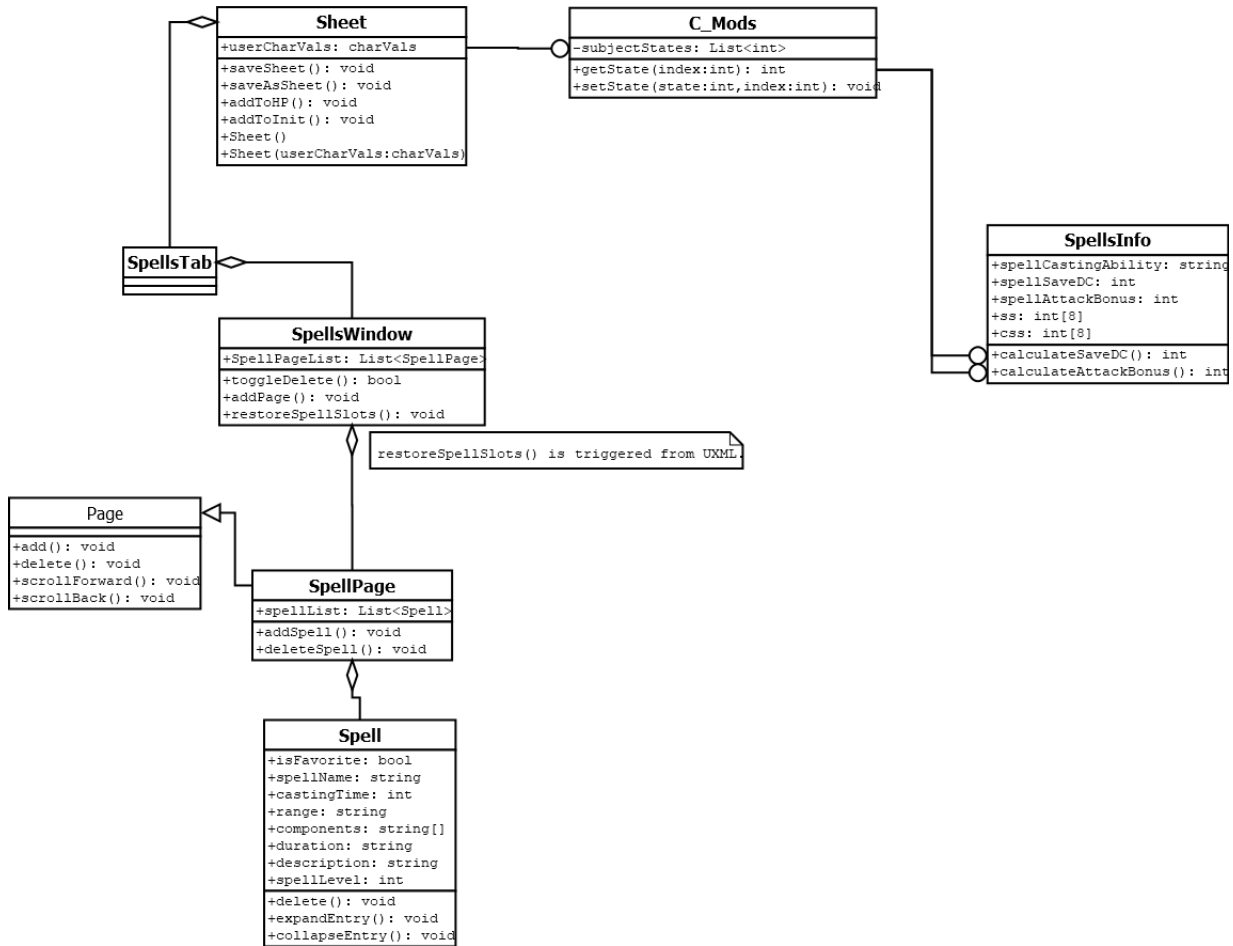


Figure 20. UML for the Spells Tab window.

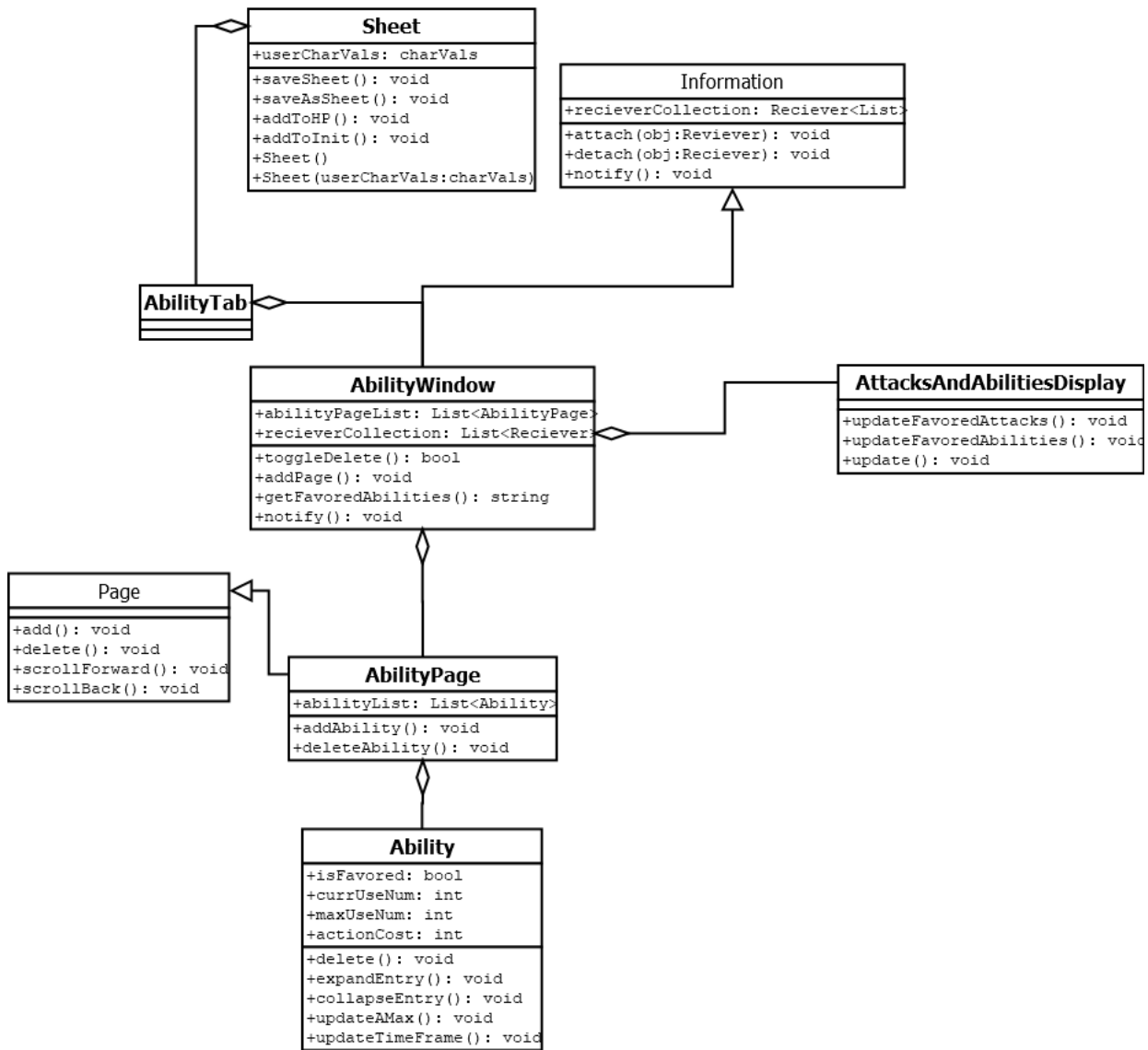


Figure 21. UML for the Abilities Tab window.

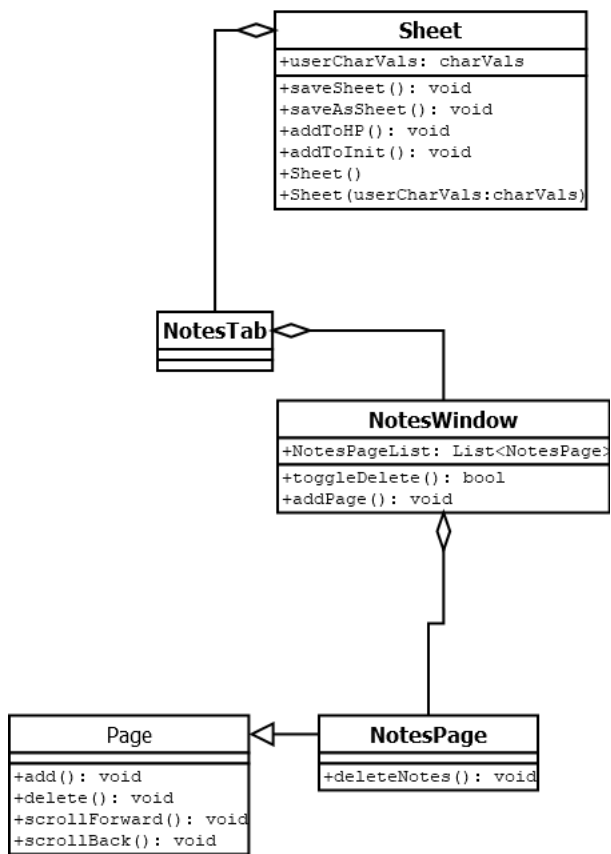


Figure 22. UML for the Notes Tab window.