Project Issues Log

A project of any size will encounter many decisions that the team has to make. If the project runs long enough, it becomes difficult to remember which alternatives were discussed, and why those alternatives may have been rejected. In extreme cases, important issues come up repeatedly, often with little or no **new** discussion simply because no one can authoritatively remember what was said when, or by whom. For your Projects in CECS 323, I am going to require that you maintain an issue log to capture the team’s discussions along the way.

For each issue, capture the following information:

1. **Issue number**: This is essentially a surrogate key for the issue. Just assign them sequentially as new issues get opened.
2. **Issue author:** The team member who raised the issue originally.
3. **Issue date**: When the issue first appeared.
4. **Issue Name**: A brief description of the issue.
5. **Issue Description**:A narrative description of the issue.
6. **Issue status**:One of the following values:
   1. **Closed**: The team has decided on a particular option.
   2. **Open**: Discussion is still open on the issue and anyone on the team can contribute new options or new arguments pro/con regarding an existing option.
   3. **Reopened**:The issue had been closed at one point but reopened because someone in the team had a significant new option to offer, and/or new arguments pro/con one or more of the existing options. **Note** a closed issue is not reopened unless the team decides that the new option(s) and/or argument(s) are significant.
7. **Options**: Each option is a possible solution to the issue. For each option, catpure:
   1. **Name:** An identifier for this option within this issue.
   2. **Option date**: The date on which the option was suggested.
   3. **Option author**:The name of the team member who suggested the option.
   4. **Option arguments**: Each argument is either in favor of, or against a particular option for a particular issue. The argument must indicate the risk/disadvantage of using that option for the given issue (a con argument) or the risk reduction/advantage of using that option for the given issue. Any references to back up the author’s arguments will be included in the arguments.
8. **Resolution**:If the issue is closed: the option that was adopted.
9. **Close date**:The date on which the team resolved the issue.

An example follows:

**Issue Number:** 1

**Issue Date:** 05/31/2021

**Issue Author:** Dave Brown

**Issue Name**: To GUI or not to GUI?

**Issue Description:** A graphical user interface (GUI) front end to a Java SE application provides the user with an easy to use, structured interface to the application data.

**Options:**

**Just use the system console.**

**Option Date:** 06/02/2021

**Option Author:** Tony Stark

**Option Arguments:**

**Pros:**

1. Using the system console for interacting with the user is easy and quick.
2. Every platform can support this approach.
3. This application is not supposed to win a beauty context, instead, it is supposed to provide us with some basic experience with the Java Persistence API.

**Cons:**

1. Using the system console is ugly and unprofessional.
2. Writing the code necessary to re-prompt the user when they make a bad entry is tedious and error prone.
3. The user is more likely to make mistakes that do not get caught right away, thus increasing their frustration with the application.

**Use a GUI framework like Java FX**

**Option Date:** 06/03/2021

**Option Author:** Dr. David Banner

**Option Arguments:**

**Pros**:

1. A well-done GUI makes the application satisfying and easy to use, which will encourage more people to give it a try.
2. These are far more fun to code than a clunky old console application.

**Cons:**

1. This will take much longer.
2. It is not actually the focus of this assignment.
3. We are not going to get any additional points for doing it with a GUI.
4. Not everyone on the team has meaningful experience in the same, or even any GUI framework.
5. Using a GUI framework increases the technology risk of the project as a whole.

**Resolution:**

Given that the summer term flies by so fast, and that there are plenty of other aspects of this project that are high risk and poorly understood, it seemed wise to stick with what we know and go with the console I/O option.