Individual Project Document

Andrew Woll

## Requirements

Epics and User Stories

As a student, I want to record my time usage so that I can keep track of time usage.

User Stories:

* Keeping track of my schedule is key to me completing my homework and projects on time.
* Keeping track of a schedule will increase the likelihood that I succeed in my plans.

Related Features:

* Entries for specific tasks will be made with their relevant information and be stored in a database to allow these entries to be queried at a later date.

As a user, I want to type “today 09:30 10:30 ‘studied Java’ :STUDY”(DATE FROM TO TASK TAG format) as an input to the application so I can make a database record and store it in the database.

User Stories:

* I would like to make entries for specific tasks I need to complete and store them so I may query them at a later date.

Related Features:

* Entries will be made to the database with the (DATE FROM TO TASK TAG) format with a date, a starting and ending time, the type of task, and associated tag for the entry.

As a user, I want flexibility in how I specify the date by using the form 2022/09/23 and being able to add AM or PM to the FROM or TO sections of a command because I want to specify elements relevant to keeping track of my schedule.

User Stories:

* I would like the capability to track tasks in a logical way based on standard time keeping formats.

Related Features:

* Based on the format of (DATE FROM TO TASK TAG), a date will be specified in the form of YEAR/MONTH/DAY and the FROM and TO sections will allow for AM and PM to be specified.

As a user, I want to query my time usage from a database by specifying a date, task, or tag related to specific entries because it gives me a way to display useful information for keeping track of my schedule.

User Stories:

* Searching for entries based on a variety of needs is important so I can see what I need to complete on a specific day and what are all the tasks I must complete for a specific project or type of task.

Related Features:

* A way to specify you are querying an entry in the a database with a specific format that allows for a date, task, or tag to be specified.

As a user, I care little about what programming language or database is used because all I am looking for is functionality.

User Stories:

* My needs are basic and solely based around having a functioning command-line task tracking application.

## Prototype

1. What is the data structure for this application, and why?

Entries will be created as strings of information to be parsed and separated into specific fields based on how they are inputted within the format. In making a query request, these strings will be parsed based on the fields in which they were stored in the database. By parsing this information, queried entries can be stored in a list of arrays which can be iterated over to display the information.

2. What is the database structure to store the information?

Each entry in a database will be stored as a collection of fields to house the information given when creating an entry.

3. What modules and interfaces do I need to make?

Classes will be needed for the functions of creating a query, submitting a query request, and displaying relevant entries to the query. Upon submitting a query request, its information must be parsed into separate variables or array indexes. In this way, information can be stored one-by-one into their specified fields in the database. Making a query request involves detecting what is being queried (i.e. dates, tasks, or tags). Based off of what is retrieved from the database due to the query, entries must have a way to be grouped and displayed upon the user interface. The algorithm retrieving entries from the database will feed the information into a list which will be utilized by another class meant for formatting this information for being displayed.

4. What are all the actions taking place within the user interface?

Creating an entry, making a query request, and displaying queried entries.

5. What are all the actions taking place within the database?

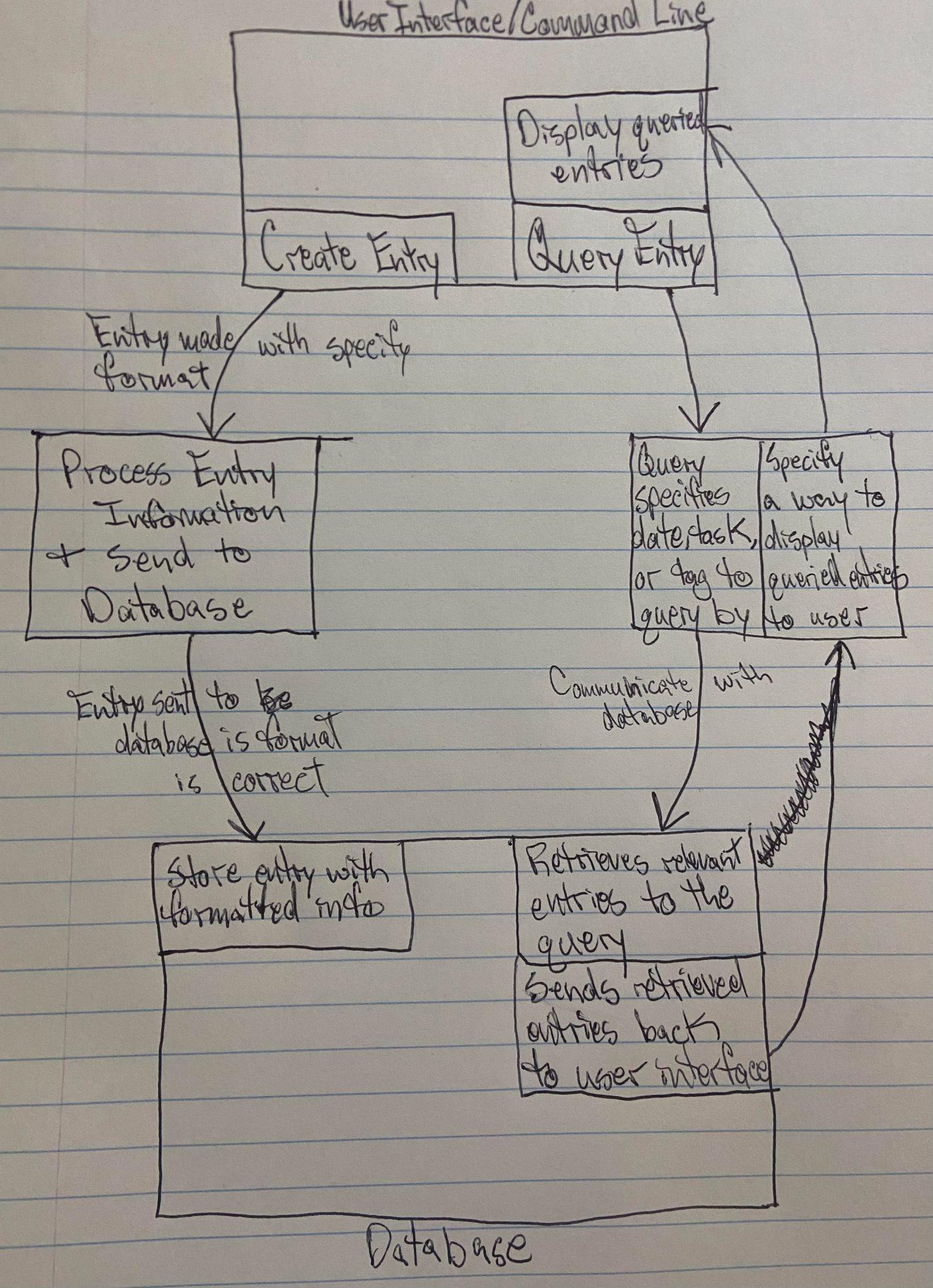
Storing an entry, requesting entries, and sending entries to the user interface.

6. What if the incorrect format is specified in creating an entry?

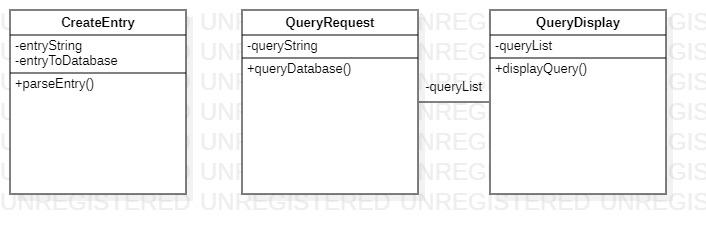
Send errors to the user and state how to specify correct queries.

7. How will query searches be displayed to the user?

Iterate over a list of arrays with indexes specific to the field in which they were stored in the database and display arrays/entries line-by-line.



### Design and UMLs



### Implementation and Unit-Tests

CreateEntry Test: specify an entry within a string and call parseEntry() to confirm it is stored in the database as expected.

QueryRequest & QueryDisplay Test: specify an query within a string and confirm that QueryDisplay displays the correct information within the user interface.

### Redesigning, Refactoring, and Revision Tools

Additional Requirements:

As a user, I want the ability to request a report of all the activities I did between a range of dates.

User Stories:

* I’m interested in seeing what kind of activities I spend my time on to understand what tasks I do the most or search for task/s I completed within a specific time period.

Related Features:

* The command ‘report (FROM TO)’, with dates being specified in the latter two sections, will display a list of tasks completed within the date range to the user.

As a user, I would like to see what activities I spend most of my time on.

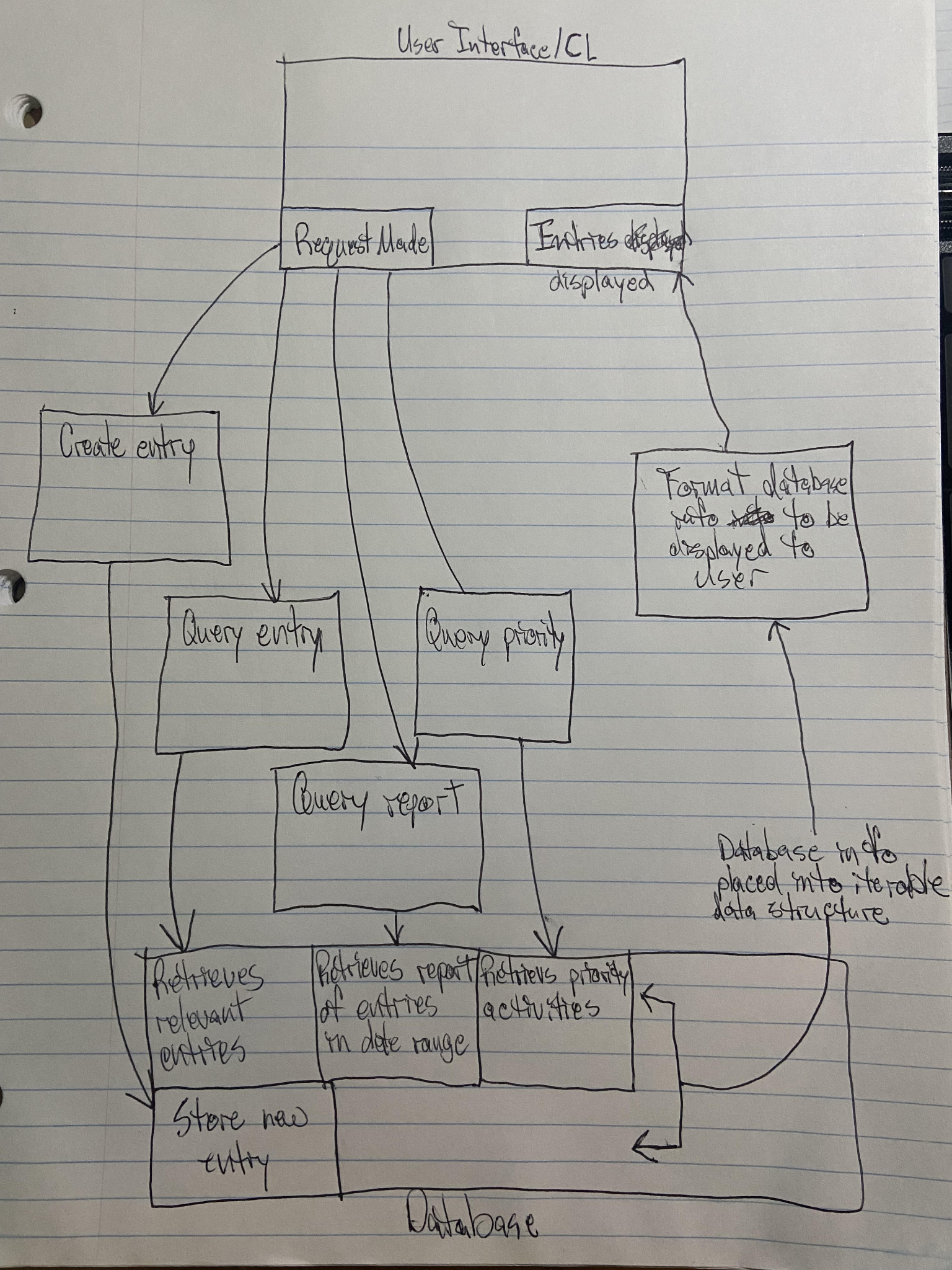
User Stories:

* Knowing what activities I spend most of my time on may give me information useful to completing these specific tasks more efficiently or at an earlier date.

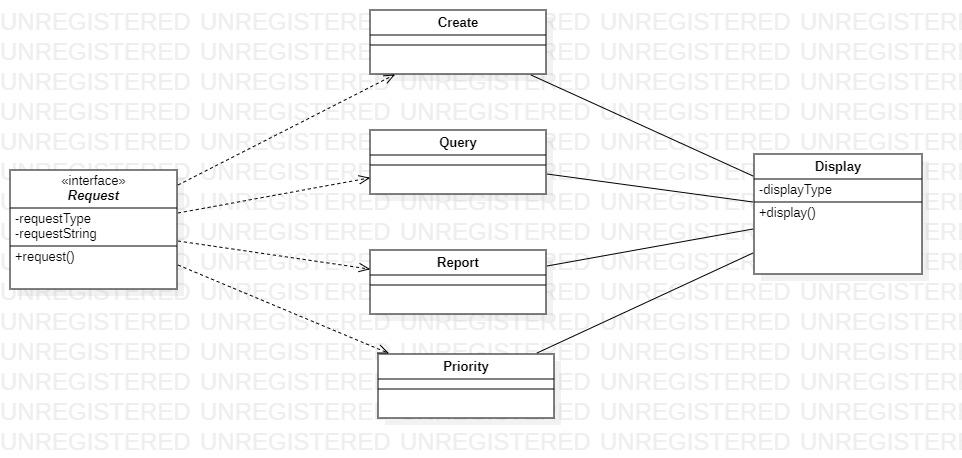
Prototype:

How will the design structure change? Will I need interfaces?

With additional actions being added to the application, it makes sense to implement interfaces for making a request at the command line (i.e. creating an entry, querying entries, etc.) and displaying information relevant to the request made. This interfaces will be implemented into classes designed to recognize what is being requested and how to display the requested information.



Design and UML:



Implementation and Unit-Tests:

Request Implemented Classes Test: test Create, Query, Report, and Priority classes with their specified requestType values. In doing so, each class will test and utilize the Display class to return a specific output to the user determined by the requestType.

### Plans

Milestone: Develop Prototype

Deadline: 10/27

Goal: create needed data structures and algorithms to procedurally complete all tasks

Milestone: Design software architecture

Deadline: 10/30

Goal: create classes and interfaces for needed actions of requesting and displaying information and utilize OOP concepts to create a functioning application

Milestone: Test application’s functionality

Deadline: 11/2

Goal: create unit tests and confirm functionality

Milestone: Refactor and design

Deadline: 11/9

Goal: sense code smells and SOLID principle violations to refactor code into its final version

Milestone: Final testing

Deadline: 11/16

Goal: refactor unit tests to confirm functionality of final version of application