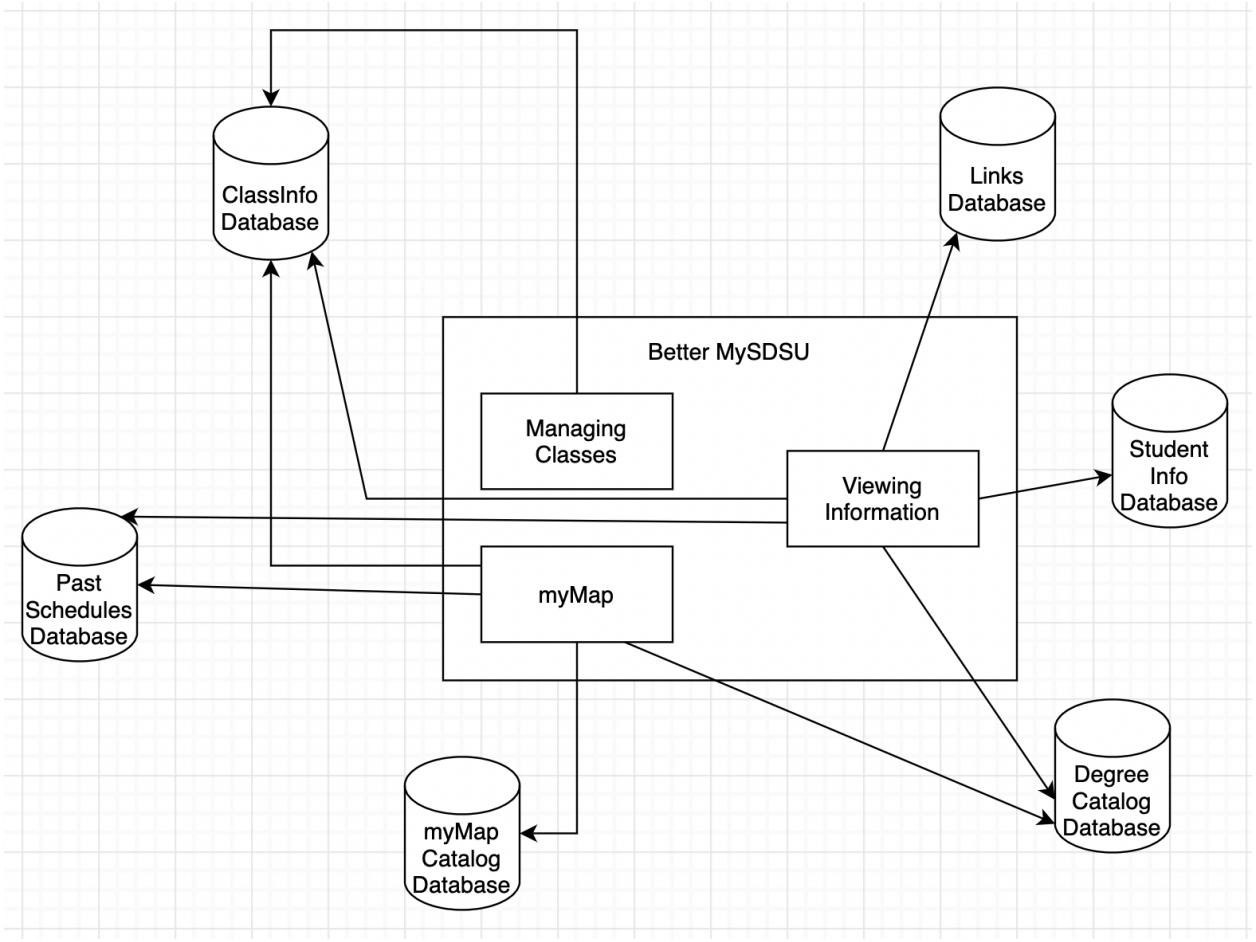
Software Design 2.0

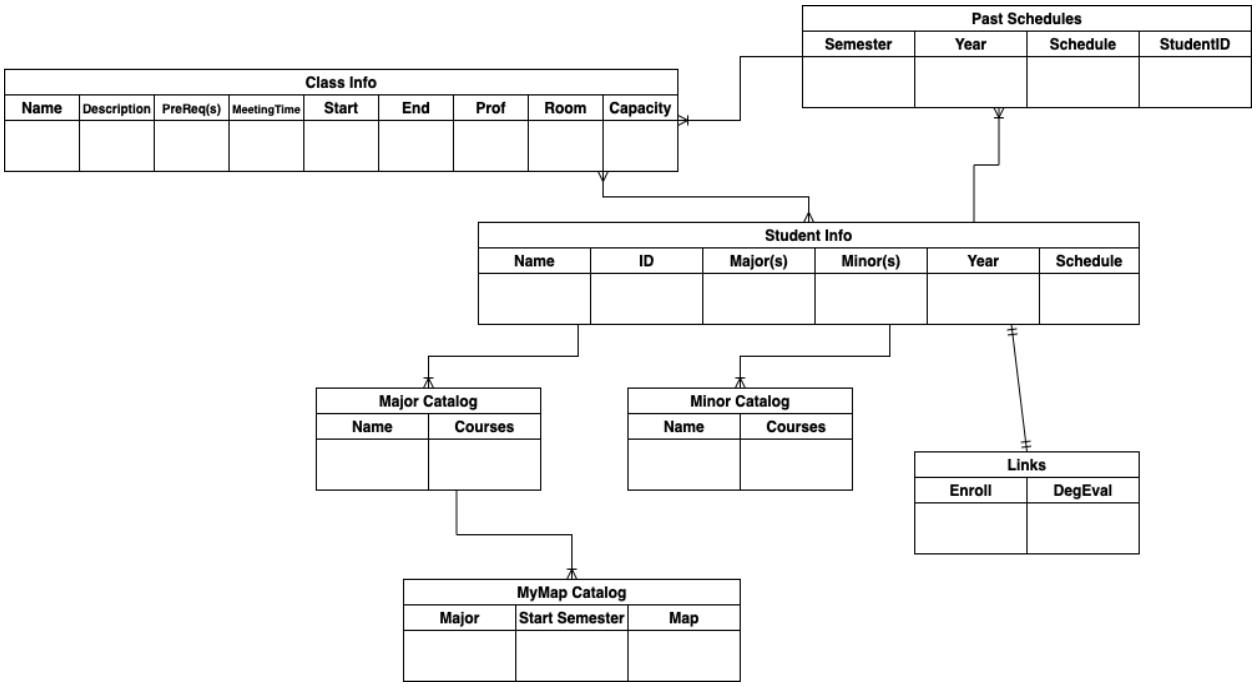
Quincy Kapsner
Sophie Krivonosov
Rage Dizon

April 13, 2023

Software Architecture Diagram



Data Management Strategy



Description

The PastSchedules class is connected to ClassInfo with a one to many relationship. One schedule goes to many classes. Then, one student can go to many schedules. Additionally, a student can have many classes. One student can also have multiple majors and minors, and the major can have multiple myMap catalogs. Finally, the student connects to links in a one to one relationship.

StudentInfo

- Name - Student's name
- ID - Student's REDID
- Major(s) - List of the student's majors (can be empty)
- Minor(s) - List of the student's majors (can be empty)
- Year - Current year (20XX)
- Semester - Current semester (ie: Fall, Spring, Summer)
- Schedule - List of ClassInfo's that are the student's classes for the current semester (can be empty)
    - Multiple ClassInfo's to each StudentInfo

PastSchedules

- StudentID - The ID of the student whose past schedules the user is accessing
    - One StudentInfo to multiple PastSchedules
- Semester - The semester the user is requesting the past schedule for

- Year - The year the user is requesting the past schedule for
- Schedule - List of ClassInfo's that are the student's classes for that semester/year
    - Multiple ClassInfo's to each PastSchedules

MyMapCatalog

- Major - The major for the MyMap
    - Multiple MyMapCatalog's to each DegreeCatalog
- StartSemester - The semester the MyMap will start at (there are different MyMaps for different starting semesters)
- Map - The MyMap based on the specified major and starting semester

Major Catalog

- Name - Name(s) of the major(s)
- Courses - Dataset of courses pertaining to the major

Minor Catalog

- Name - Name(s) of the minor(s)
- Courses - Dataset of courses pertaining to the minor

ClassInfo

- Name - Name of the class
- Description - Description of the class
- PreReq(s) - All Preqreqs for a particular course
- MeetingTime - Meeting times for the course
- Start - Start date for the course
- End - End date for the course
- Prof - Professor teaching the course
- Room - Room number of the course
- Capacity - Max amount of students allowed in a course

Links

- Enroll - Links for individual enrollment appointments
- DegEval - Links for tailor made degree evals depending on the student

Design Decisions

- *How many databases you chose and why.*
    - We chose to use six databases because we felt each database was unique enough to warrant its separate existence. There is no overlapping information across the databases we made and each field is necessary for the functionality of the system.
- *How you split up the data logically.*

    - We essentially started with the databases that we had already established in the UML diagram, and we then kept making subdivisions based on if they were needed or not
- *What are possible alternatives you could have used (both in technology and organization of data)?*
    - A possible alternative could have been building custom NoSQL databases for the MyMap catalog or Student past and current schedules.
- *What are the tradeoffs between your choice and alternatives?*
    - The choice provided us with separate databases with no overlapping information, and we were able to ensure that each field was necessary for the system to work the best. The alternative of NoSQL could provide agility and more availability, but it's more inconsistent with its data retrieval. In general, the SQL system provides faster processing and improves the functionality of the system.