CS848F26 Project Proposal - Ahmed El Shatshat

Proposal Title: Survey and Meta-Analysis of Database Design Methods Using Principles of Requirements Engineering

Description: As the methods for data collection, management, and analysis diversify, so too do the considerations behind choosing a database provider, or in designing a custom-built database. While database design was largely standardized in the days of Web 2.0, to mixed result, today the domain of database system technology is once again highly diversified to meet the demand of developers. A one-size-fits-all solution has become naive.

In the typical case, a software project will have some use case that requires the integration of a database into the computer-based system. Developers must be aware of what to consider when evaluating a candidate database. This effectively boils down to comparing their system's requirements for a database against the requirements of a database themselves. This problem has two main facets: developing requirements and models for databases themselves, and a deriving a computer-based system's requirements for a database. These two facets must be aligned in order for the project to continue.

However, despite the database needs of developers maturing, it appears the methodology of picking the right database for your computer-based system did not mature with it. A regimented approach for selecting the appropriate database could thus help developers solve this problem with increased ease. To that end, this project seeks to perform a survey of database design and database selection methods, and then perform a meta-analysis of these methods through the lens of a requirements engineer.

This project will seek to address the following research questions:

RQ1: What are the current methods for designing database systems, available in literature?

RQ2: What are the current methods for choosing an appropriate database system for one's computer-based system, available in literature?

RQ3: How well do these methods compliment each other? What are the differences in pathology between them?

RQ4: How do these methods hold up against best practices within requirements engineering?