Software Modeling and Process

Simon Codrington III

Grand Canyon University

SWE-520: Advanced Software Engineering Fundamentals

Dr. Wibbenmeyer

April 17, 2024

Activity 5.5: Sequence Diagram of Student Registering for Class

The diagram below shows a sequence diagram that outlines the process of a student registering for a class. There are some things that do not need to be accounted for at this stage, such as where and how to determine if the class is too full. Due to this I decided to give this functionality to the school database. This is the class status that is being sent back from the database. A certain type of SQL query will be needed to determine this from the school catalog object. I decided it was best to use the registry system to determine if the student is even able to register for classes. If not, there is no need to even see classes. This is done after logging in and if they can log in and are enrolled, then they can register for class. Finally, the school database is leveraged to determine if the class is full. Most databases do not do this so the catalog object will use the database data payload to determine if the class is available. (Sommerville, 2015 pg 133).

A diagram of a project

Description automatically generated

Activity 5.7 ATM Activity Diagram

The diagram below is an activity diagram for the process of an ATM system. This is meant for high level context and to show the migration of data. When the user prompts the system to enter the PIN, that is the only system outside the ATM that it needs to communicate with. This can easily be done with a third-party API for an ATM is an IOT device.

A diagram of a flowchart

Description automatically generated

Activity 10.1: 6 Needs for Dependable Software

In a sociotechnical society, the need for dependable software is paramount. This dependability is the foundation that these societies are based on. My background and study in sociology has given me a foundation of understanding of how these agents in society interact with the societal system. In a sociotechnical society, a lot of technology is meant to be used in tandem with humans. (Sommerville, 2015 pg 278) Humans are keenly adept at solving problems. This brings its own sense of dependability and technology needs to be able to keep up with its human counterpart. Reason number to speaks to the competitive nature of the market. When systems go down companies lose money and miss deadlines and such. Thirdly, is the topic of trust. Companies spend a lot of time garnering trust with their users and employees. Unreliable systems will disrupt this trust. Regulation and compliance are the next reasons. A lot of nonfunctional requirements of the system is to make sure that the company is in compliant with legal stakeholders. System interruption or non-dependable security would cause ruin in this domain. The most important reason is safety. (Sommerville, 2015 pg 278) A lot of systems are real time systems and are crucial for maintaining human life. I can not imagine being in a plane and the generators are reliant on some API and then the API goes down causing the plane to fall out of the sky. The final point speaks to efficiency. This is the main reason for western societies to switch to a tech dependent one. Loss in dependability is loss in efficiency, which is the point of all of this.

Activity 10.2 Most Important Factors to Dependable Systems

When it comes to dependability the most important factors are availability, reliability safety, security and resilience (Sommerville, 2015 pg 275). Availability is self-explanatory. To be dependable, you must be available. In software this usually means 24/7. Reliability speaks to accuracy. The system must give the user the expected outcome upon use. Safety is about how the system will affect the environment and potentially cause damage. (Sommerville, 2015 pg 275) Security is the system of being able to protect again intentional and accidental threats. And resiliency means that can stand up to all of the issues that arise in a sociotechnical society. This means being able to service power outages, user error, threat attacks, or any other events that could cause a disturbance.

Activity 10.6 Dependable Processes

If one wants to build and architect dependable software, then the process of creating said software had better be dependable as well. I tell my daughter “Do it right, then do it fast”. It’s my way of teaching her to do it right the first time. This way you do not have to waste time to go back to fix things. In software engineering this will manifest in some key characteristics. A dependable process has key characteristics. The first of which is auditable, this means that outside parties of the process can check to see if standards are being adhered to. (Sommerville, 2015 pg 284) Firms do this for consultation. Diversity speaks to having multiple checks and redundant verification processes. (Sommerville, 2015 pg 284) Documentable means that we can model the process and each activity has a document that is produced. Robustness is how recoverable from failures the process is, and diversity and redundancy help with this. And finally standardized, which means there are the presence of coding best practices, and documentation for staging and production (Sommerville, 2015 pg 284)..

*Citations*

Sommerville, I (2015). *Software engineering* (10th ed.). Pearson.