Software Architecture and Design

Simon Codrington III

Grand Canyon University

SWE-520: Advanced Software Engineering Fundamentals

Dr. Wibbenmeyer

April 24, 2024

Activity 6.7 Layered Architecture for Asset Management System of a Utility Company

Proposed Layered Architecture:

A close-up of a white box

Description automatically generated

(Sommerville, 2015, p.163)

The above figure is my proposed high level conceptual and structural view of an asset management system intended for a power company. This business level application has been split into four separate layers (or five if you include the REST API) for maintainability and easy changes. The separations of concern also help to secure the system from outside threats and make the system more robust.

The first layer of the system is the database. In this case, the power company has its own database, therefore we are confined to any nonfunctional requirements of said system and how to make transactions. For the sake of discussion, we will say the database is a SQL database housed in Azure which is a cloud platform. From there an API will be built using .Net Web API framework housed in it own project. This will handle any and all request to perform all CRUD operations (create, replace, update, delete).

The next layer is the use case / information retrieval layer. Here in its own project a class library can be created as a data processing repository. Here all the use case logic for adding and asset, updating an asset, getting information about an asset, and more will be contained.

The following layer is the client management system layer. Here we will implement the log in and authentication system. This can be done by building a separate API just for authentication and communicating with the database. This will also allow us to verify the type of device the user is login in on. With this information we can give the appropriate UI to the appropriate user for data entry is mostly done on the mobile device where information viewing is not.

The final layer is the user interface layer built using .Net Blazor Hybrid. This is a relatively new framework offered by .Net that is a mix between the web framework Blazor and the mobile and desktop framework MAUI. This is especially pertinent due to the company needing two different interfaces for the system and two completely different uses for the system. The people in the field enter asset data using the mobile device and everyone else uses it for information viewing. The people in the field need a specific user experience that is tailored towards said device and vice versa.

Activity 7.1 Reconfigure and Report Status Use Case for Weather Station

|  |  |
| --- | --- |
| System | Weather Station |
| Use Case | Reconfigure System |
| Actors | Weather Station, Control System |
| Data | When the system for the weather station sends a reconfigure message to the control system, it will take the input of the reconfiguration settings and then send those settings to the weather station and turn on the system to the running state. |
| Stimulus | The weather station sends request over network to the control system |
| Response | The weather station is put to a running state with the new configurations entered |
| Comments | This can only occur when the system is in a shutdown state |

|  |  |
| --- | --- |
| System | Weather Station |
| Use Case | Report Status |
| Actors | Weather Station, Weather Information System |
| Data | The system for the weather station will report its status to the information center for testing. |
| Stimulus | The weather station sends request over network to the information system |
| Response | The data sent from the weather station is tested by the information system and then sent back to the system once test is complete |
| Comments | This can only occur when the system is in a running state |

Some of the context for the weather system was taken from the state management diagrams and the use case diagram for who the actors were (Sommerville, 2015, p.193). While it did not explicitly say it, the control system is what is expected to help with the state management. Based on the diagram, certain actions only occurred during certain states of the system such as the system being shut down. The state architecture also spoke to the testing of the system when the weather station sends a status report.

Activity 7.3 Multi System UML Design

* Cell Phone Messaging System

A diagram of a user

Description automatically generated

* Printer for Personal Computer A screenshot of a computer

  Description automatically generated
* Personal Music System

A screenshot of a computer program

Description automatically generated

* Bank Account

A screenshot of a computer

Description automatically generated

* Library Catalog

A screenshot of a computer

Description automatically generated

(Sommerville, 2015, p.193).

*Citations*

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