Graduate Team 3

Project Title: Using Model Discovery on an ASP.NET application to provide understanding of the systems structure to the Maintenance Team

CS 5340 Software Maintenance
Semester: Spring 2025
Semester Project Final Presentation
Team members: John Williams

Permission is granted for publication on the UCCS website.

Professor: Dr. Armin Moin
Teaching Assistant (TA): Ariful Rabbani

Agenda

- **Semester Project**
- **Research**
- **Live Demo**
- **Q&A**

Project Overview

- License Asset Management Systems (LAMs)
  - ***Allows users to purchase software license for specific applications.***
  - ***Allows developers to embed license requirements in their applications.***
- The customer: The maintenance team of the License Asset Management System.
- The end user: Users of Software that is license controlled by the License Asset Management System.
- The problem being addressed:
  - ***There is a lack of understanding of the systems structure.***
  - ***There is a lack of confidence that the system meets the requirements.***
- The value being added: A better understanding of the systems structure and increase in confidence that the systems meets the requirements.

Work breakdown

ID	Description	Status	Owner	Completion Date
M-1	Review context and use case diagrams	Completed	John	03/12/25
M-2	Unit test and document Navigation	Completed	John	3/12/25
M-3	Unit test and document the Cart. Refactor Blazor implementation.	Completed	John	4/14/25
M-4	Unit test and document Administration	In Progress	John	4/7/25
M-5	Unit test and document Security and Deployment	In Progress	John	4/21/25

https://uccs-team-cs5320.atlassian.net/jira/software/projects/SCRUM/boards/1

Open-source Software License

- The GitHub repository contains a README.md file with important project information.
- Project README
- It is also included as part of this package.

Copyright 2024-2025 Johnny C. King and John L Williams Jr

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

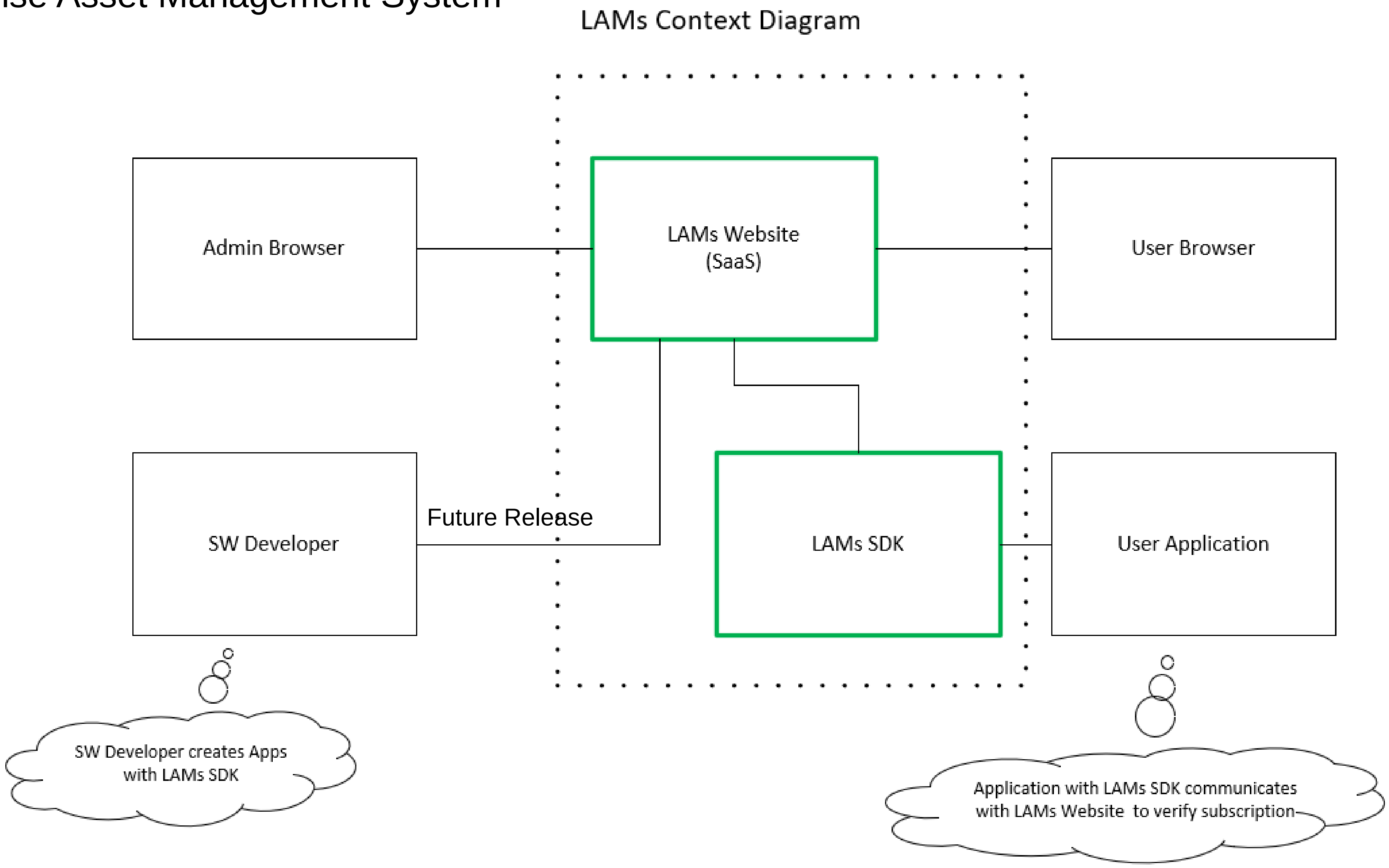
This license is also controlled by the Manning Publisher License, who provided a great deal of guidance and source material in the creation of the application. Please read the details in the link provided below.

Manning Publisher License

Open-source Project GitHub Repository

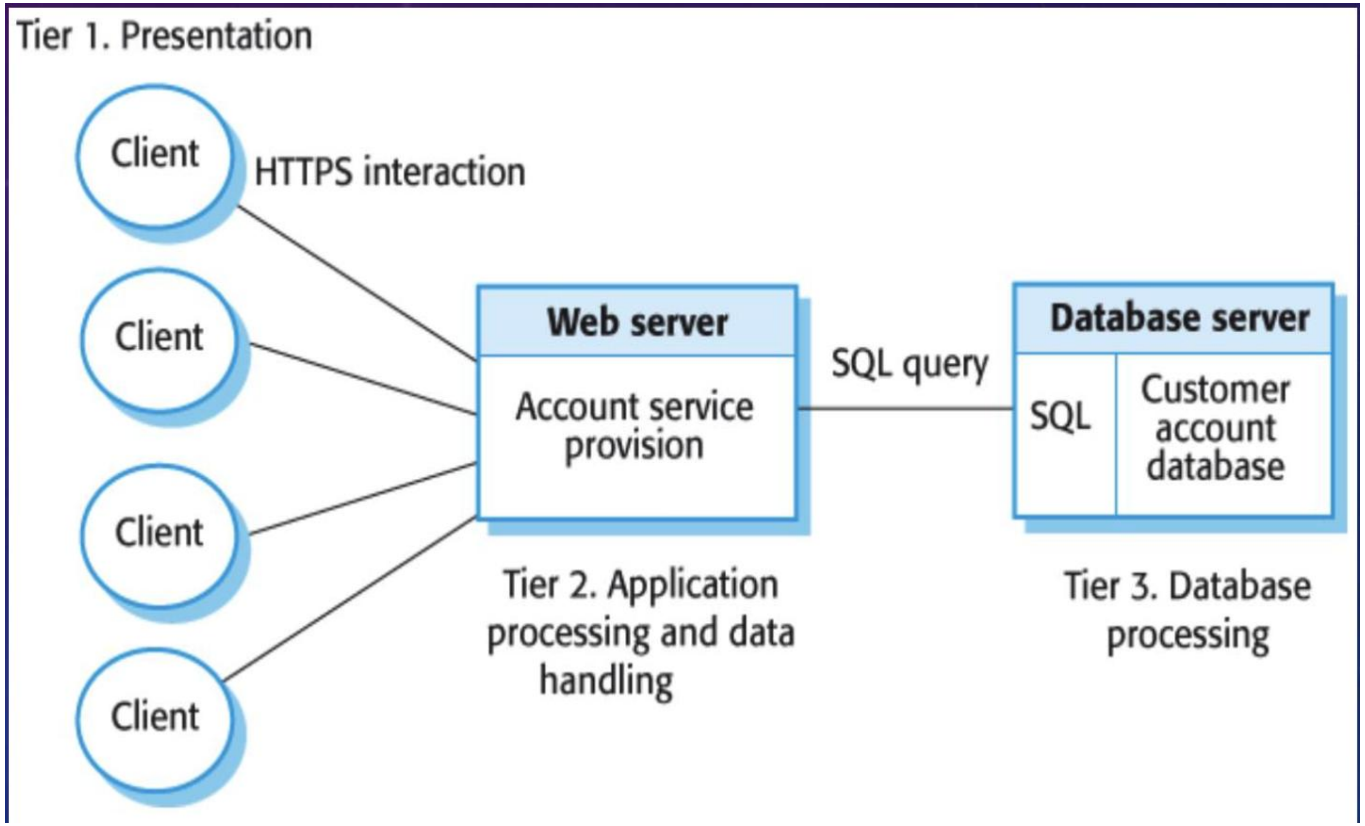
- Project GitHub Link

LAMs – License Asset Management System



Multi-tier Client Server Architectural Pattern

- **Good for large volume of transactions**
- **Able to scale as more clients sign up for the service it will be better for larger scale**
- **Good for applications where data is volatile**



Tier 1. Presentation
Client
Client
Client
Client
HTTPS interaction
Web server
Account service provision
SQL query
Database server
SQL
Customer account database
Tier 2. Application processing and data handling
Tier 3. Database processing

Types of Work

- **Enhancements**
  - – Refactor Blazor implementation
- **Perfective**
  - – Add Unit Test
  - – Add UML Diagrams

Relevance to Classroom Instruction

- **Context Diagram**
- **UML Use Case Diagrams**
  - – Updated to show the systems required features.
  - – Helped create a mental model of the system to increase program comprehension.
- **UML Class Diagrams**
  - – Generated using Enterprise Architect
- **Unit Test**
  - – Added unit test to demonstrate that the system meets the requirements.

Relevance to Classroom Instruction

- **Component Test (System Test – Unit Test)**
  - Example: Test Viewing the Stores Products
- **System Testing**
- **Automated Testing**
  - Used Visual Studio Test Explorer to automate testing
  - Helped ensure that updates did not break the system
- **Activity based Software Maintenance**
  - Enhancement to refactor Blazor implementation without changing the requirements

Relevance to Classroom Instruction

- **Evidence Based Software Maintenance**
  - Documentation (Updative)
    - ***UML Class Diagram Generation using Enterprise Architect***
  - Automated Testing (Preventative and Perfective)

Research

Using tools for model discovery and maintenance

Background

- **Legacy systems without UML class diagrams are common.**
- **Software Maintenance Engineers have struggled to understand these complex legacy systems.**
- **Vendors have been working on providing tools that allow the discovery of models within these legacy systems for decades to help engineers understand the structure and behavior of these systems.**

Some Existing Tools for Model Discovery

- **Visual Studio 2022**
- **Enterprise Architect**
- **MoDisco**

Experimental Results

- **Visual Studio 2022**
  - Capable of generating UML Class diagrams one file at a time
- **Enterprise Architect**
  - Capable of generating UML Class diagrams for an entire Visual Studio Solution or Project
- **MoDisco**
  - Capabilities not tested

Evaluation

- **Enterprise Architect is very fast at performing model discovery of an entire Visual Studio Solution in one step.**
- **Model discovery of ASP.NET applications typically took less than one minute using Enterprise Architect.**
- **Model discovery of the same ASP.NET application using Visual Studio tool approximately fifteen minutes due to its one-at-a-time process.**

Summary

- **Model Discovery is a great way to increase the understanding of a legacy system that does not include UML documentation.**
- **There are existing tools that help facilitate model discovery. Enterprise Architect does a great job at processing an entire Visual Studio Solution with one step.**

Future Work

- **Finding ways to extract other types of models, such as activity diagrams for methods and complex properties.**
- **Using model discovery in forward engineering methodologies for reengineering legacy systems.**

Live Demo

Questions and Answers

Thank You!

John Williams – jwilli11@uccs.edu

UCCS University of Colorado Colorado Springs

CU University of Colorado

Boulder | Colorado Springs | Denver | Anschutz Medical Campus