

# Open Source .NET Blog

DESIGN DOCUMENT

34

Cylosoft

Selim Mohamed

PJ DeBisschop, Cole Mulligan, Sebastian Kazun, Thien Nguyen

[sdmay20-34@iastate.edu](mailto:sdmay20-34@iastate.edu)

[sdmay20-34.sd.ece.iastate.edu](http://sdmay20-34.sd.ece.iastate.edu)

Revised: 9/30 v1

# Executive Summary

## Development Standards & Practices Used

List all standard circuit, hardware, software practices used in this project. List all the Engineering standards that apply to this project that were considered.

## Summary of Requirements

- ASP.NET 4.8
- MSSQL
- Bootstrap 4.3
- CKEditor 4
- Microsoft Identity
- CRUD Blog Posts
- Comments (Approve / Deny)
- Categories (Filterable)
- Tags (Filterable)
- Site settings (Change theme)
- View most recent posts

## Applicable Courses from Iowa State University Curriculum

- COM S 309 Software Development Practices
- COM S 363 Intro to Database Management
- COM S 319 Construction of User Interfaces

## New Skills/Knowledge acquired that was not taught in courses

- .NET 4.8 Framework
- Microsoft SQL
- C#
- Bootstrap 4.3
- Microsoft Identity/Owin Authentication
- Model/View/Controller Architecture

# Table of Contents

1 Introduction	4
1.1 Acknowledgement	4
1.2 Problem and Project Statement	4
1.3 Operational Environment	4
1.4 Requirements	4
1.5 Intended Users and Uses	4
1.6 Assumptions and Limitations	5
1.7 Expected End Product and Deliverables	5
2. Specifications and Analysis	5
2.1 Proposed Design	5
2.2 Design Analysis	6
2.3 Development Process	6
2.4 Design Plan	6
3. Statement of Work	6
3.1 Previous Work And Literature	6
3.2 Technology Considerations	7
3.3 Task Decomposition	7
3.4 Possible Risks And Risk Management	7
3.5 Project Proposed Milestones and Evaluation Criteria	7
3.6 Project Tracking Procedures	7
3.7 Expected Results and Validation	7
4. Project Timeline, Estimated Resources, and Challenges	8
4.1 Project Timeline	8
4.2 Feasibility Assessment	8
4.3 Personnel Effort Requirements	8
4.4 Other Resource Requirements	8
4.5 Financial Requirements	9
5. Testing and Implementation	9
5.1 Interface Specifications	9
5.2 Hardware and software	9

5.3	Functional Testing	9
5.4	Non-Functional Testing	9
5.5	Process	10
5.6	Results	10
6.	Closing Material	10
6.1	Conclusion	10
6.2	References	10
6.3	Appendices	10

**List of figures/tables/symbols/definitions** (This should be similar to the project plan)

Figure 1: System Block Diagram

Figure 2: User Flow Diagram

# 1 Introduction

## 1.1 ACKNOWLEDGEMENT

Thank you to Cylosoft for their technical and design support throughout the project.

## 1.2 PROBLEM AND PROJECT STATEMENT

### General Problem Statement

- In the current state of the world there are a limited amount of open source .NET blogs for users to take advantage of. Those that exist are outdated and fall behind today's technology. Because of this Cylosoft would like their own blog platform which include additional features and allow them more control.

### General Solution Approach

- A simple to use blog builder with an admin area to control content for readers.

## 1.3 OPERATIONAL ENVIRONMENT

The ASP.NET server and database will be hosted through Microsoft's Azure cloud service.

## 1.4 REQUIREMENTS

### Functional Requirements

Normal User Area for viewing content

- A blog visitor shall be able to sort posts by most recent
- A blog visitor shall be able to filter posts by tags
- A blog visitor shall be able to filter posts by categories
- A blog user shall be able to view approved comments on posts
- A blog visitor shall be able to sign up/log in for an account
- A blog user shall be able to comment on posts
- A blog visitor shall be able to search for specific posts

Admin Area for controlling content

- An admin shall be able to Create, Read, Update, and Delete posts
- An admin shall be able to view new comments and approve them for others to see
- An admin shall be able to add posts to certain categories for users to filter by
- An admin shall be able to edit user roles
- Ad admin shall be able to change the theme

## Non-Functional Requirements

- The software shall be accessible at any time
- The software shall not crash
- The software shall properly handle errors behind the scenes
- The software shall display content properly based on size and device
- The software shall utilize Microsoft Identity for security
- The software shall be designed in a way that allows easy changes to the codebase

## UI Requirements

- The UI must be visually pleasing
- The UI shall be able to change theme

## 1.5 INTENDED USERS AND USES

Our intention is to build an Open Source .NET blogging platform that allows Cylosoft to keep their customers informed about their activities. While also being flexible enough for other individuals or businesses to use as their own blog.

## 1.6 ASSUMPTIONS AND LIMITATIONS

### Assumptions

- The end product will be open source.
- The end product will be able to support multiple users at a time.
- The end product will be viewable on various media devices

### Limitations

- The length of time to produce the end product will not surpass May 2020.
- Enough database space to hold thousands of blog posts
- The cost of the product cannot exceed \$0

## 1.7 EXPECTED END PRODUCT AND DELIVERABLES

- Open Source .NET Blog Platform with source code
- Documentation
  - project plan, design document, database design, architecture design

Delivery date: May 1, 2020

## 2. Specifications and Analysis

### 2.1 PROPOSED DESIGN

Our project will be built using .NET, so we have been experimenting with .NET web application development. Our design will use a properly managed and optimized database.

### 2.2 DESIGN ANALYSIS

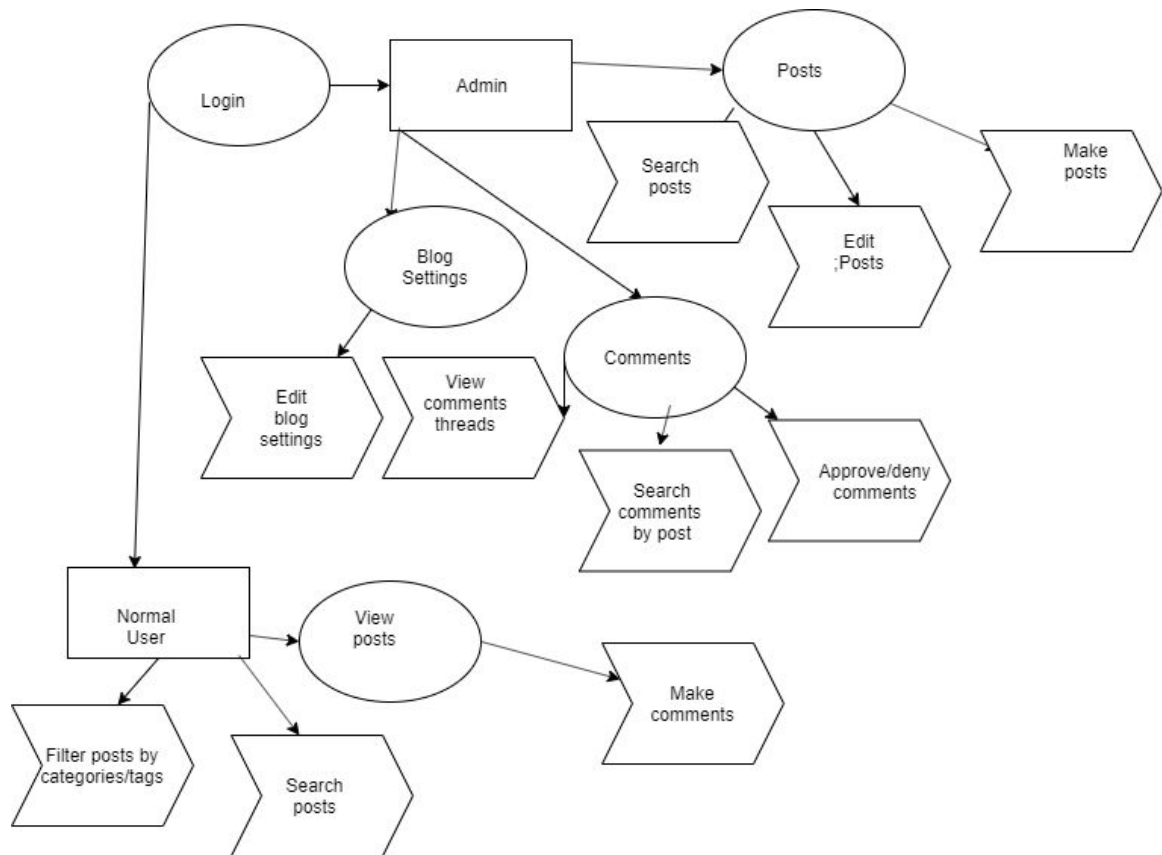


Figure 2

So far, we have began mocking up our user processes and researching .NET development. We plan to begin work with the User Flow Diagram in Figure 2 in mind. These are most of the requirements for our users to be able to operate their blogs and where in the website they can do each of these functions.

what we did so far and did it work

observations, thoughts, ideas

Strengths

Weaknesses

### 2.3 DEVELOPMENT PROCESS

Our team will be following the Agile Development Process. We believe that adhering to the 12 principles will help immensely, allowing us to work flexibly and keep us on track. We will be using Trello as our agile board to help us assign tasks to the team. This will allow us to have a visual indicator on our progress for our current sprint and the effort of each teammate. Separating tasks into sprints allows us to more easily plan our work and make it manageable. We plan to release our first prototype by the end of the semester in December. From there we will work incrementally to further develop features and refine our project. We will end with a final product in early May.

### 2.4 DESIGN PLAN

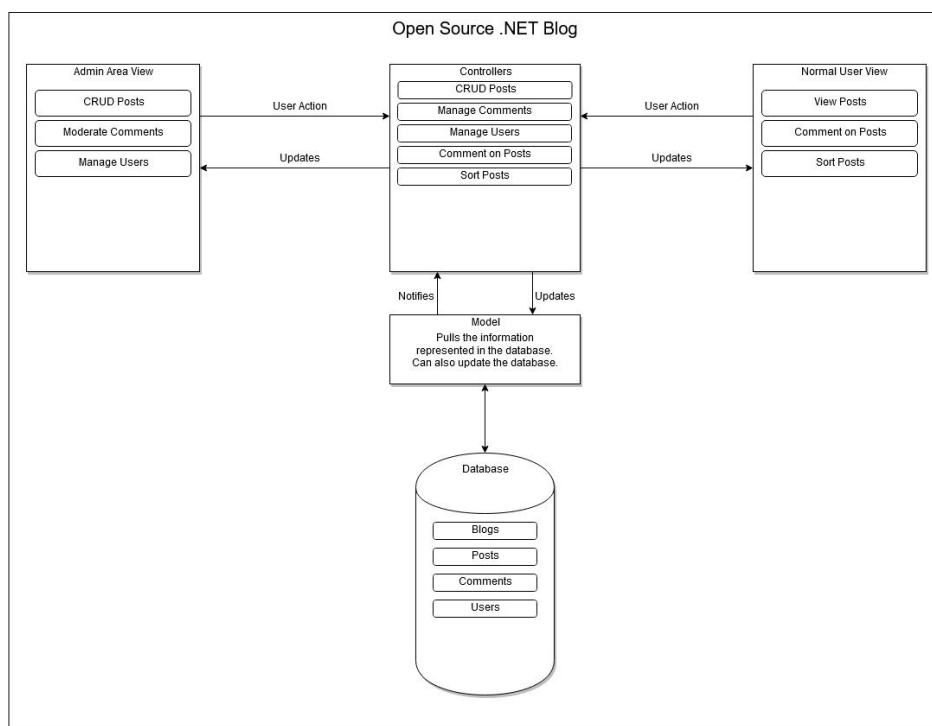


Figure 1

This is our first conceptual block diagram for how our Model View Controller system will function. You can start in either the Admin Area View on the left or Normal User View on



the right, but both follow the same flow. Starting from the Admin Area View you will have options like CRUD Posts, a functionality allowing you to Create, Update, Read, or Delete any and all posts on the blog. After deciding what action you would like to perform the VIEW sends a user action request to the controller module. Inside this main module are also others pertaining to different functionalities. Since we are doing CRUD Posts from the view we will also use the corresponding controller. This controller then updates the model, the model being our representation of the data contained inside the database. The model, after receiving the update, changing whatever needs to be changed will then notify the controller which will then update the corresponding view.

### 3. Statement of Work

#### 3.1 PREVIOUS WORK AND LITERATURE

Include relevant background/literature review for the project

- If similar products exist in the market, describe what has already been done
- If you are following previous work, cite that and discuss the **advantages/shortcomings**
- Note that while you are not expected to “compete” with other existing products / research groups, you should be able to differentiate your project from what is available

Detail any similar products or research done on this topic previously. Please cite your sources and include them in your references. All figures must be captioned and referenced in your text.

#### 3.2 TECHNOLOGY CONSIDERATIONS

Highlight the strengths, weakness, and trade-offs made in technology available.

Discuss possible solutions and design alternatives

#### 3.3 TASK DECOMPOSITION

In order to solve the problem at hand, it helps to decompose it into multiple tasks and to understand interdependence among tasks.

#### 3.4 POSSIBLE RISKS AND RISK MANAGEMENT

Include any concerns or details that may slow or hinder your plan as it is now. These may include anything to do with costs, materials, equipment, knowledge of area, accuracy issues, etc.

### 3.5 PROJECT PROPOSED MILESTONES AND EVALUATION CRITERIA

What are some key milestones in your proposed project? Consider developing task-wise milestones. What tests will your group perform to confirm it works?

### 3.6 PROJECT TRACKING PROCEDURES

What will your group use to track progress throughout the course of this and next semester?

### 3.7 EXPECTED RESULTS AND VALIDATION

What is the desired outcome?

How will you confirm that your solutions work at a **High level**?

## 4. Project Timeline, Estimated Resources, and Challenges

### 4.1 PROJECT TIMELINE

- A realistic, well-planned schedule is an essential component of every well-planned project
- Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity
- A detailed schedule is needed as a part of the plan:
  - Start with a Gantt chart showing the tasks (that you developed in 3.3) and associated subtasks versus the proposed project calendar. The Gantt chart shall be referenced and summarized in the text.
  - Annotate the Gantt chart with when each project deliverable will be delivered
- Completely compatible with an Agile development cycle if that's your thing

How would you plan for the project to be completed in two semesters? Represent with appropriate charts and tables or other means.

Make sure to include at least a couple paragraphs discussing the timeline and why it is being proposed. Include details that distinguish between design details for present project version and later stages of project.

#### 4.2 FEASIBILITY ASSESSMENT

Realistic projection of what the project will be. State foreseen challenges of the project.

#### 4.3 PERSONNEL EFFORT REQUIREMENTS

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be based on the projected effort required to perform the task correctly and not just “X” hours per week for the number of weeks that the task is active

#### 4.4 OTHER RESOURCE REQUIREMENTS

Identify the other resources aside from financial, such as parts and materials that are required to conduct the project.

#### 4.5 FINANCIAL REQUIREMENTS

If relevant, include the total financial resources required to conduct the project.

## 5. Testing and Implementation

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)
2. Define the individual items to be tested
3. Define, design, and develop the actual test cases
4. Determine the anticipated test results for each test case
5. Perform the actual tests
6. Evaluate the actual test results
7. Make the necessary changes to the product being tested
8. Perform any necessary

retesting

9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you've determined.

### 5.1 INTERFACE SPECIFICATIONS

- Discuss any hardware/software interfacing that you are working on for testing your project

### 5.2 HARDWARE AND SOFTWARE

- Indicate any hardware and/or software used in the testing phase
- Provide brief, simple introductions for each to explain the usefulness of each

### 5.3 FUNCTIONAL TESTING

Examples include unit, integration, system, acceptance testing

### 5.4 NON-FUNCTIONAL TESTING

Testing for performance, security, usability, compatibility

### 5.5 PROCESS

- Explain how each method indicated in Section 2 was tested
- Flow diagram of the process if applicable (should be for most projects)

### 5.6 RESULTS

- List and explain any and all results obtained so far during the testing phase
  - - Include failures and successes
  - - Explain what you learned and how you are planning to change it as you progress with your project
  - - If you are including figures, please include captions and cite it in the text
- This part will likely need to be refined in your 492 semester where the majority of the implementation and testing work will take place
- Modeling and Simulation:** This could be logic analyzation, waveform outputs, block testing. 3D model renders, modeling graphs.
- List the **implementation Issues and Challenges.**

## 6. Closing Material

### 6.1 CONCLUSION

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

### 6.2 REFERENCES

This will likely be different than in project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).

### 6.3 APPENDICES

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc. PCB testing issues etc. Software bugs etc.