

12-19-2022

## Hybrid Life Cycles In Software Development

Eric Vincent Schoenborn

Follow this and additional works at: <https://scholarworks.gvsu.edu/gradprojects>



Part of the [Computer and Systems Architecture Commons](#), [Databases and Information Systems Commons](#), [Data Storage Systems Commons](#), [Graphics and Human Computer Interfaces Commons](#), [Software Engineering Commons](#), and the [Systems Architecture Commons](#)

---

### ScholarWorks Citation

Schoenborn, Eric Vincent, "Hybrid Life Cycles In Software Development" (2022). *Culminating Experience Projects*. 215.

<https://scholarworks.gvsu.edu/gradprojects/215>

This Project is brought to you for free and open access by the Graduate Research and Creative Practice at ScholarWorks@GVSU. It has been accepted for inclusion in Culminating Experience Projects by an authorized administrator of ScholarWorks@GVSU. For more information, please contact [scholarworks@gvsu.edu](mailto:scholarworks@gvsu.edu).

Hybrid Life Cycles in Software Development

Eric Vincent Schoenborn

A Project Submitted to

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Applied Computer Science

School of Computing and Information Systems

April 2023



The signatures of the individuals below indicate that they have read and approved the project of Eric Vincent Schoenborn in partial fulfillment of the requirements for the degree of Master of Science in Applied Computer Science.

---

<name of project advisor>, Project Advisor	Date
--	------

---

<name of GPD>, Graduate Program Director	Date
--	------

---

<name of unit head>, Unit head	Date
--------------------------------	------

## **Abstract**

This project applied software specification gathering, architecture, work planning, and development to a real-world development effort for a local business. This project began with a feasibility meeting with the owner of Zeal Aerial Fitness. After feasibility was assessed the intended users, needed functionality, and expected user restrictions were identified with the stakeholders. A hybrid software lifecycle was selected to allow a focus on base functionality up front followed by an iterative development of expectations of the stakeholders. I was able to create various specification diagrams that express the end projects goals to both developers and non-tech individuals using a standard architectural design. Of the 99 initially specified goals 73 have been completed due to setbacks with refactoring, time underestimates, unknown/unplanned issues, and scope change.

## **Introduction**

For my project, I created a web application for a local gym named Zeal Aerial Fitness. Building an application for a company, rather than creating one for myself, was a key part of this project. Because the desired outcome was not my own, it allowed me to show my skill in requirement elicitation and architecture.

The goal of this project was to create a plan based off another person's requests and then use that plan to create an application. The plan would consist of high-level requirements and use cases, along with low level architectural designs. The result would need to be both understandable to all stakeholders involved while containing enough detail that any competent software developer could create this software solution.

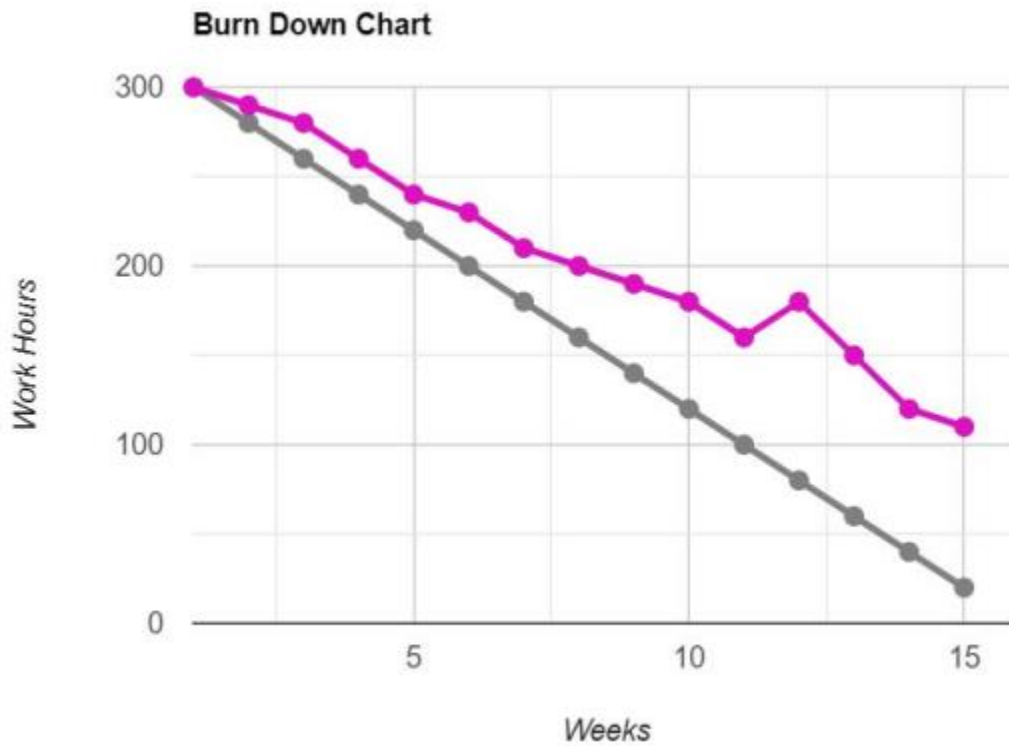
The project has an included code base. However, the purpose of the code is not to show my skills as a developer. Rather, the code base is meant to prove that my requirement specification and architecture plans are not just customer friendly, they are also able to be used to write functional software.

I got my undergraduate degree in computer science at GVSU and have been working professionally as a software developer since. My goal for my masters is to move forward in my career and assume the role as requirements engineer, software architect, or team lead. I believe that the classes I have completed and this project show that I am ready for these roles.

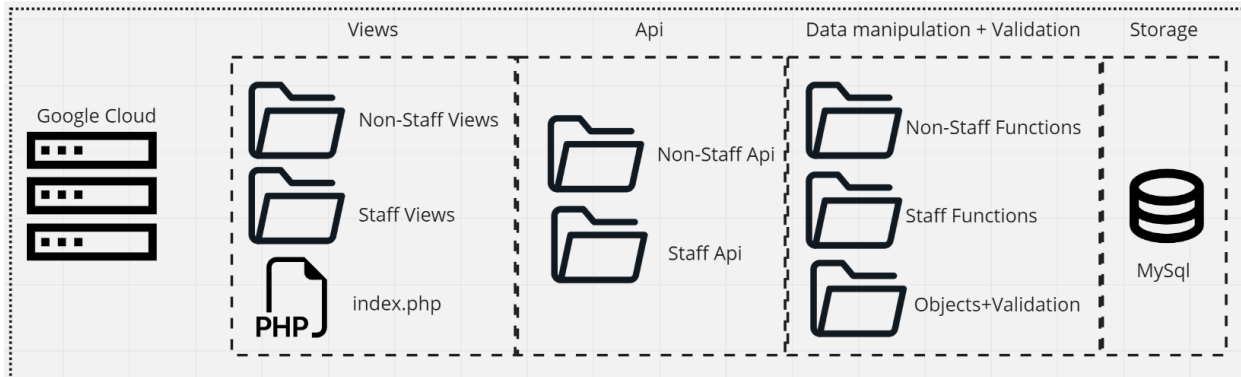
# Project Management

For this project, I chose to implement a Hybrid life cycle. The hybrid I chose started with the Waterfall lifecycle and then transitioned into Scrum. The reason this life cycle was effective was it allowed me to get much of the non-user testable code, such as the database design and API, completed with a minimal site presentation done without having stakeholder meetings in between. The first iteration had a little over half the requirements in a testable state with all the background functionality and architecture in place and vetted.

From this point forward I switch to a Scrum approach where I only focused on two use cases at a time. One of which had been created in the waterfall portion that needed updates, and one that was new development. This setup allowed me to keep a constant supply of completed use cases to be tested without overwhelming my workload or the stakeholders.



# Organization



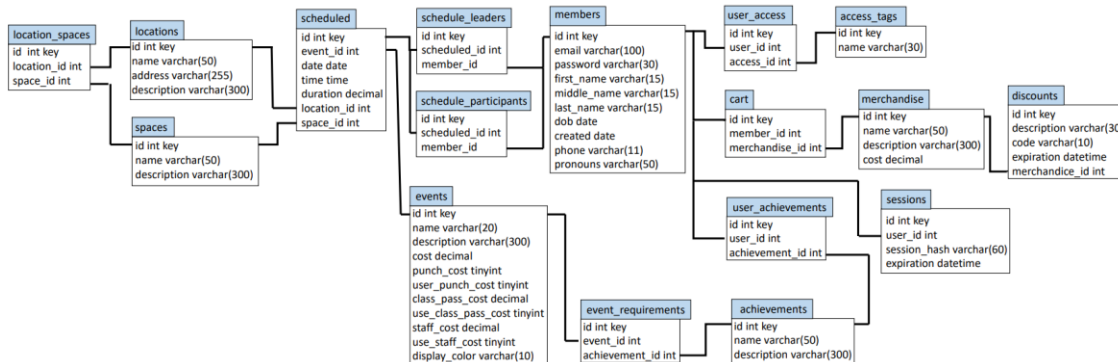
The web application was deployed on virtual machine hosted on google could. I chose the cloud serveries because I have had previous experience with it, and it is relatively inexpensive. The application was divided into four major sections, with most of those sections have a subsection to divide staff-based functionality from non-staff-based functionality.

The first section is the Views section. It holds the PHP and HTML code to allow users to view entity data and interact with it. The main view is the index.php file and servs as the landing page for this web application. From there, with the correct permissions, a user can navigate to the rest of the pages.

The next section is the API section. The API section decouples the frontend from the backend (section 1 from section 3). Because of this, it would be easy to replace any of the four sections with a completely new solution without needing to update much of the code. Additionally, this allows for future technologies to be integrated with this web application. The owner of Zeal mentioned that they would also want to add a phone app and kiosk to this solution in our second meeting.

The third section is the backend. This is the code responsible for data validation, error handling and data flow. Though, there is some data validation in the first section, the final checks will be one after the API section. Because of this, we are ensured that any data coming from current or future devises will be held to the same standard of correctness.

The final section is the data storage section. For this project I chose to use a MySQL databases. Along with being a free tool, MySQL databases written in 'normal form' are easily extensible. Additionally, MySQL (and Sql) are a common development practice, making this solution easy maintainable



## Reflection

Out of the 99 requirements, I was able to complete 73 of them. There were the usual setbacks in the form of under estimation of tasks. Additionally, I chose to develop in vim to save on cost and I have realized how much I take the tools from Visual Studio for granted. I have gotten too used to my IDE underlining syntax errors and highlighting logic failed lines during runtime. However, I consider this a good learning experience overall and believe it made me a better developer. In the end, the largest setback I had was in the form of scope creep. The reason I wanted to go with a real company instead of a made-up project is because it is very hard to simulate real life problems and changes in plans. About three quarters way through this project the owner told me that they would be expanding by adding another location. This meant I would need to add in locations and spaces management tools so that the application and users would not get confused on which event would be referenced. Though it was frustrating to have to rewrite some code in the “done pile” and add more to the workload, I am glad it happened. This single event strengthens my argument for using a real company for my Master’s project.



## **Conclusions**

Though this project has almost all the requirements completed, it still has months of development left. This is due to site appearance. Just because all the requirements are satisfied does not mean that all the stakeholders are. Additionally, there will probably be additional requirements added as the full web application goes online for user testing. The current plan is to back off on development for the next semester so that I can graduate on time. Then, ramp development back up and have version 1 of the site open to the public by the end of the summer. If the stakeholders approve of my work and ask me to stay with them, I will then begin development on the Zeal Aerial Fitness phone app.

## Appendices

All source code can be found here: <https://github.com/ericschoenborn/zealaf>

To run this web application locally, you will need to create you own database using the queries in `zealaf/mysql/tables.sql`. You will also need to update the last few lines in `zealaf/mysql/requests.php` to have the server, name, user, and password for your new table.