Formal Technical Inspection – Austin Gray

Planr, an Agile Project Planning Application

Version 2.0

Submitted in partial fulfillment of the requirements of the degree of Master of Software Engineering

Prepared by

Michael Blakeman

CIS 895 – MSE Project

Kansas State University

Table of Contents

[1. Introduction 3](#_Toc88814407)

[2. Results 3](#_Toc88814408)

# Introduction

Below are the results from Austin Gray’s technical inspection of the requirements, architecture, and test coverage. Austin Gray is a MSE peer at KSU and can be contacted at [sid34@ksu.edu](mailto:sid34@ksu.edu).

# 2. Results

|  |  |  |
| --- | --- | --- |
| **Items to be Inspected** | **Pass / Fail / Partial** | **Comments** |
| Each symbol used in the class diagram / struct diagram are according to UML standards. | Pass | The UML diagrams all follow the UML standard and show a very well planned software architecture. |
| Each class / struct in the UML diagram have clear specifications as to their purpose in the System Architecture Document. | Pass | Each UML diagram and component shows a clear understanding of the design laid out in the System Architecture Document. The plans were easy to follow and did not have any unnecessary bloat. The plan leverages dependencies well, e.g. Realm. |
| All requirements in the Software Requirements Specification (located in the Vision Document) have been covered in the Test Plan Document. | Pass | Functional Requirements are met, both the UML design and code reflect this. Much of the requirement outline the type of data to encase with features, engineers, etc and this data is captured and surfaced in Planr |

Hey Mike,

I have filled out the table above to reflect my findings on Planr. Overall, I think this is a great project. As someone who deals with an agile workflow every day, I can say that the simplicity and efficiency of how the data is surfaced is a nice change. We use Azure DevOps and the complexity really takes away from the efficiency of the Agile paradigm. I think that your system architecture is simple yet robust and appears to be easily extendable if the need arises. Your code quality throughout the project seems solid. I think this project showcases a firm knowledge about all aspects of the software development life cycle. Thank you for the opportunity to review this and I hope to hear about the results of the course!

Happy Thanksgiving,

Austin Gray