Kristianstad University
SE-291 88 Kristianstad
Sweden
+46 44 250 30 00
www.hkr.se

Pineapple Planner

Agile Development Project Report

Max Sellick, Varvara Aladyina, Deinoras Krasauskas, Azhaf Kahn, Simon Ostini

March 2025

Title
Pineapple Planner

Subtitle
Agile Development Project Report

Programme
Software Development

Authors
Max Sellick, Varvara Aladyina, Deinoras Krasauskas, Azhaf Khan, Simon Ostini

Keywords
Agile, Scrum, Project idea

Contents

1 Introduction

¡Instructions: Write an introduction to your application and its purpose. Also include screenshot¿

1.1 Restrictions

¡ Instructions: List any restrictions that you think is worth mentioning.¿

1.2 Enhancements

¡ Instructions: Explain possible extra features that you plan to implement into your application. If you have no extra features, you can remove this 1.2 section.¿

2 Requirements

3 Design and Implementation

¡ Instructions: Describe your design in this chapter. List one class per sub chapter and add some simple class diagrams to illustrate relations (inheritance and/or associations) between the main classes.¿

The application implementation is far to large to list and explain every single class. As the application is built with the object oriented language C# almost everything is a class.

3.1 Class1

¡Description of this class. ¿

3.2 Class2

4 Test Results

Table 2 below contains the current status of implemented and tested requirements. ¡Instructions: This table shall map 1-1 to the table in Chapter 2. The test result for each requirement shall be one of the following: NOT IMPLEMENTED, PASSED or FAILED.¿

5 Summary and Conclusion

This chapter contains a summary and conclusion of the work that was carried out in this project as well as reflections and thoughts about working methods and challenges.

5.1 Weekly Progress

Below is a short summary of what was done each week.

5.1.1 Week 1

¡Instructions: Describe what you did this week. You can see it as a developer's weekly diary. Try to answer the following questions: What did you do this week? Did you meet any challenges? What was difficult? Did you get stuck with something? What went well and what went bad? How were your time estimates? Did you overestimate or underestimate the time of some features? Did your priorities seem OK? What have you learned during this week? Get help of individual reflections on Canvas.¿

5.1.2 Week 2

5.1.3 Week 3

5.1.4 Week 4

5.2 Difficulties and challenges

Below is a list of notable challenges that came up during this project and that took a long time to solve.

5.2.1 Blazor WPF wrapper

One significant challenge we faced several times was the fact that we decided to implement our Blazor SSR web application within a WPF wrapper to be ran as a desktop application. Having a web based application is very comfortable for development, especially when it comes to building user interfaces. Also you can always fallback to use JavaScript as the app runs in the browser. On the other hand, having a WPF wrapper limited us when implementing certain features.
For instance, redirects that were necessary for the authentication with Firebase turned out to be rather difficult.

5.2.2 Dependency Injection issues

Blazor and WPF both have their own dependency injection frameworks and ensuring proper integration between the two required some trial and error. Some services that worked seamlessly within Blazor did not behave as expected when instantiated inside the WPF wrapper. For example, singleton services shared between Blazor and WPF sometimes led to unintended issues. The whole services setup took quite some time.

5.2.3 Lack of console outputs for debugging

Another major difficulty we encountered was the lack of direct console output when running the Blazor application within the WPF wrapper. Usually, web applications benefit from browser developer tools, in particular the console outputs. However, within the WPF environment, there was neither a .NET console nor a browser console available making debugging significantly harder. We often ended up rendering output to the user interface for testing.

5.2.4 Firebase API Quota Exceeded

During development and testing, we occasionally exceeded Firebase's API quota. This typically happened when we testing extensively in a short period or when our

application accidentally fell into an endless request loop. We had to be extremely careful not to get stuck in infinite render loops. Otherwise we had to wait one day for the firebase cooldown to continue development.

5.3 Correctness of time estimates

¡Instructions: Look back on your time estimates and discuss your results. How accurate were they? What have you learned about time estimates and how can you get better in next project?¿

5.4 Priority decisions

¡Instructions: Look back on your feature priority settings. Did you prioritize the right features? Did you succeed to deliver the highest prioritized features? Did you disagree with the examiner on some features? Have you learned anything about setting priorities?¿

5.5 Conclusion

¡Instructions: Look back on the whole project. Here you can write a bit more freely about your thoughts on this project. What was your overall experience? How was the teamwork? What did you learn? Can you list some points that you will do better in next project? Other thoughts. ¿

6 References