



Kristianstad
University
Sweden

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	4
1.1	Restrictions	4
1.2	Enhancements	4
2	Requirements	4
3	Design and Implementation	4
3.1	Class1	4
3.2	Class2	4
4	Test Results	4
5	Summary and Conclusion	5
5.1	Weekly Progress	5
5.1.1	Week 1	5
5.1.2	Week 2	5
5.1.3	Week 3	5
5.1.4	Week 4	5
5.2	Difficulties and challenges	5
5.2.1	Name of Challenge/Difficulty 1	5
5.3	Correctness of time estimates	5
5.4	Priority decisions	6
5.5	Conclusion	6
6	References	7

1 Introduction

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

PineapplePlanner is a windows application that functions as a to-do list in a calender where a person creates a task with a specific start and finish date, selects a priority if needed (high, medium, low) then gives the task a name and submits it. The purpose of PineapplePlanner is to help users make and plan their schedules.

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.

The possible extra features we planned to implement or that has been implemented was custom usernames for users which was already implemented, dark-mode option in settings which transforms the application to dark-mode for users who would prefer to have it that way. AI was another feature that got implemented to help users create tasks faster. Not yet implemented features were Notification features in settings that would allow users to enable/disable notifications, Language option in settings for users whom do not understand english or would rather prefer have it in another language.

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.

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.

Week 1: Set up the project github, jira, firewall, create authentication, refactored code.

5.1.2 Week 2

Week 2: Drawn and implemented UI, implemented UI, implemented email verification, created and implemented logo, implemented calendar, settings, to-do list.

5.1.3 Week 3

Week 3: Added unit tests, fix minor mistakes, implemented dark mode, implemented tags.

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 Name of Challenge/Difficulty 1

Instructions: List the most difficult tasks in this project and describe why they were difficult. Did you learn something, e.g. how to handle very difficult programming problems?

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