

Projektdokumentation

Projektname: ProductivityGarden

Name

Jonas Huber

Matrikelnummer

IU14085128

Modul

Projekt: Software Engineering

DLMCSPSE01_D

Datum

17.01.2025

Inhalt

1. Projektbeschreibung	2
2. Installation und notwendige Ressourcen.....	3
3. Risikomanagement	4
Potenzielle Risiken.....	4
Technische Risiken.....	4
Benutzerbezogene Risiken.....	4
Sicherheitsrisiken.....	5
Projektbezogene Risiken.....	5
Bewertungstabelle.....	6
4. Zeitplanung.....	7
5. Literaturverzeichnis.....	8

Abbildungsverzeichnis

Abbildung 1 - Gantt-Diagramm.....	7
-----------------------------------	---

1. Projektbeschreibung

Im modernen Alltag sehen sich Menschen häufig mit der Herausforderung konfrontiert, sich zu konzentrieren und ihre Zeit effektiv zu managen. Studien belegen, dass Gamification einen positiven Einfluss auf Motivation und Produktivität ausübt (Matallaoui, 2016, S. 4). Gamification, auch Spielifizierung genannt, bezeichnet die Anwendung spielerischer Elemente und Mechanismen in einem nicht-spielerischen Kontext, beispielsweise im Bildungsbereich (Huseynli, 2024, S. 45). Die Integration eines flexiblen Pomodoro-Timers, eines klassischen Timers sowie einer Stoppuhr ermöglicht es den Nutzer:innen, ihre produktiven Phasen präzise zu planen und nachzuverfolgen. Dies erfolgt in einer entspannenden und spielerischen Umgebung, wodurch eine nachhaltige Steigerung der Produktivität gefördert und gleichzeitig eine Reduktion von Stress erzielt werden soll. Zur Optimierung des Zeitmanagements besteht die Möglichkeit, die aufgewendete Zeit einzelnen Projekten zuzuweisen.

Ein wesentlicher Bestandteil der Anwendung ist ein Belohnungssystem, bei dem Nutzer:innen für produktive Zeit Punkte sammeln können. Diese Punkte können dazu verwendet werden, virtuelle Gärten mit Pflanzen und Dekorationen zu gestalten. Die Nutzer:innen haben die Möglichkeit, mehrere Gärten anzulegen und individuell anzupassen. Dies fördert nicht nur die Motivation, sondern auch Entspannung und Kreativität.

Die Anwendung ist für das Betriebssystem Windows optimiert und bleibt im Hintergrund aktiv, um eine nahtlose Nutzung zu ermöglichen. Die Speicherung sämtlicher Daten erfolgt lokal, um die Privatsphäre der Nutzer:innen zu schützen. Die Oberfläche verzichtet auf ablenkende Elemente, um den Nutzer:innen eine ungestörte Konzentration auf das Spielerlebnis und die Produktivität zu ermöglichen. ProductivityGarden ist werbefrei und bietet keinen Zugang zu Paywalls oder Premium-Inhalten.

Im Gegensatz zu herkömmlichen Apps zur Prokrastinationskontrolle richtet sich ProductivityGarden an eine breite Zielgruppe und ist für alle geeignet, die ihr Zeitmanagement verbessern möchten, eine zugleich entspannende und kreative Herangehensweise.

2. Installation und notwendige Ressourcen

Um die Anwendung zum Laufen zu bekommen helfen:

1. Repository von GitHub pullen:
https://github.com/Xecu114/iu_pse
2. "README" lesen und beachten
3. Es werden zwei weitere Python-Pakete benötigt, damit die Anwendung ausgeführt werden kann:
 - a. PyQt6
 - b. Pygame

Diese können auch mit Hilfe des „install_py_libs.bat“ Skripts automatisch installiert werden.

3. Risikomanagement

Potenzielle Risiken

Bei der Risikoanalyse werden potenzielle Risiken identifiziert, deren Eintrittswahrscheinlichkeit und Schadensausmaß bewertet und geeignete Gegenmaßnahmen vorgeschlagen. Die verschiedenen Risiken werden für mehr Übersichtlichkeit in Bereiche kategorisiert.

Technische Risiken

1. Leistungsprobleme bei hoher Last oder Datenmengen
 - Beschreibung: Die Anwendung könnte bei intensiver Nutzung oder Speicherung vieler Daten langsam werden, was die Benutzerfreundlichkeit beeinträchtigt.
 - Eintrittswahrscheinlichkeit: Mittel
 - Schadensausmaß: Hoch (Nutzer:innen könnten das Programm frustriert abbrechen)
 - Gegenmaßnahmen:
 - Nutzung performanter Python-Bibliotheken und Implementierung von Optimierungsmöglichkeiten (Recherche benötigt)
 - Optimierung der Datenbankzugriffe und Minimierung redundanter Prozesse.
 - Vorab Lasttests durchführen.
2. Datenverlust durch unerwartetes Beenden der Anwendung
 - Beschreibung: Fortschritte oder Einstellungen der Nutzer:innen könnten bei einem Absturz verloren gehen.
 - Eintrittswahrscheinlichkeit: Mittel
 - Schadensausmaß: Hoch
 - Gegenmaßnahmen:
 - Regelmäßige Autosave-Funktion implementieren.
 - Backup-Mechanismen für kritische Daten einbauen.

Benutzerbezogene Risiken

3. Mangelnde Akzeptanz der Benutzeroberfläche
 - Beschreibung: Die minimalistische Oberfläche oder die Visualisierung des virtuellen Gartens könnten als unattraktiv empfunden werden.
 - Eintrittswahrscheinlichkeit: Mittel
 - Schadensausmaß: Mittel
 - Gegenmaßnahmen:
 - Feedback einholen und iterativ verbessern.
4. Fehlende Motivation zur Nutzung durch schlechte Umsetzung der Gamification
 - Beschreibung: Nutzer:innen könnten die Anwendung nicht regelmäßig verwenden, da die Gamification nicht motivierend genug ist.
 - Eintrittswahrscheinlichkeit: Mittel
 - Schadensausmaß: Hoch
 - Gegenmaßnahmen:
 - Belohnungssystem abwechslungsreich gestalten (z. B. freischaltbare Inhalte, Fortschrittsanzeige).
 - Regelmäßige Erfolgserlebnisse integrieren, die den Nutzer:innen einen Nutzen aufzeigen.

Sicherheitsrisiken

5. Datenschutzverletzungen

- Beschreibung: Speicherung personenbezogener Daten könnte Sicherheitslücken enthalten oder DSGVO-Vorgaben nicht vollständig erfüllen.
- Eintrittswahrscheinlichkeit: Niedrig
- Schadensausmaß: Sehr hoch (rechtliche Konsequenzen, Vertrauensverlust)
- Gegenmaßnahmen:
 - Speicherung ausschließlich lokal durchführen.
 - Anonyme Datenerfassung sicherstellen.
 - Code- und Sicherheits-Audits vor Veröffentlichung.

Projektbezogene Risiken

6. Zeitüberschreitung oder Nichterfüllung von Anforderungen

- Beschreibung: Das Projekt könnte mehr Zeit als geplant in Anspruch nehmen oder einige Features könnten nicht rechtzeitig fertiggestellt werden.
- Eintrittswahrscheinlichkeit: Hoch
- Schadensausmaß: Hoch
- Gegenmaßnahmen:
 - Projektplanung in Milestones aufteilen.
 - Priorisierung der wichtigsten Features vornehmen.

7. Unerwartete technische Schwierigkeiten (z. B. mit Python-Bibliotheken)

- Beschreibung: Verwendete Bibliotheken könnten Bugs enthalten oder nicht die gewünschten Funktionen bieten.
- Eintrittswahrscheinlichkeit: Mittel
- Schadensausmaß: Mittel
- Gegenmaßnahmen:
 - Vorab sorgfältige Auswahl der Bibliotheken treffen.
 - Alternativen evaluieren und Backup-Lösungen bereithalten.

Bewertungstabelle

Tabelle 1 fasst die identifizierten Risiken zusammen, indem sie diese nach Eintrittswahrscheinlichkeit und Schadensausmaß kategorisiert. Auf Basis dieser Bewertung wird die Priorität der Maßnahmen abgeleitet, um die wichtigsten Risiken gezielt zu adressieren.

Risiko	Eintritts- wahrscheinlichkeit	Schadensausmaß	Maßnahmenpriorität
Leistungsprobleme	Mittel	Hoch	Hoch
Datenverlust	Mittel	Hoch	Hoch
Mangelnde Akzeptanz der UI und Visualisierung	Mittel	Mittel	Mittel
Fehlende Nutzungsmotivation	Hoch	Hoch	Hoch
Datenschutzverletzungen	Niedrig	Sehr hoch	Hoch
Nutzer-Manipulation (Cheating)	Mittel	Mittel	Mittel
Zeitüberschreitung	Hoch	Mittel	Hoch
Technische Schwierigkeiten	Mittel	Mittel	Mittel

Tabelle 1 – Risikoanalyse-Bewertungstabelle

4. Zeitplanung

Bei der Zeitplanung sind einige Kriterien zu berücksichtigen. Zum einen ist es schwierig, den Aufwand für die einzelnen Aufgaben im Voraus abzuschätzen, zum anderen ist es schwierig abzuschätzen, wie viel Zeit an welchen Tagen zur Verfügung steht. Daher ist es wichtig, die Zeitplanung so zu gestalten, dass der aktuelle Fortschritt ständig mit den Zielen verglichen werden kann und gegebenenfalls kleine Anpassungen vorgenommen werden können.

Deswegen wird die Projektzeitplanung kontinuierlich mithilfe eines Gantt-Diagramms überwacht, das zugleich zur Fortschrittskontrolle dient. Dieses ist in Abbildung 1 zu sehen.

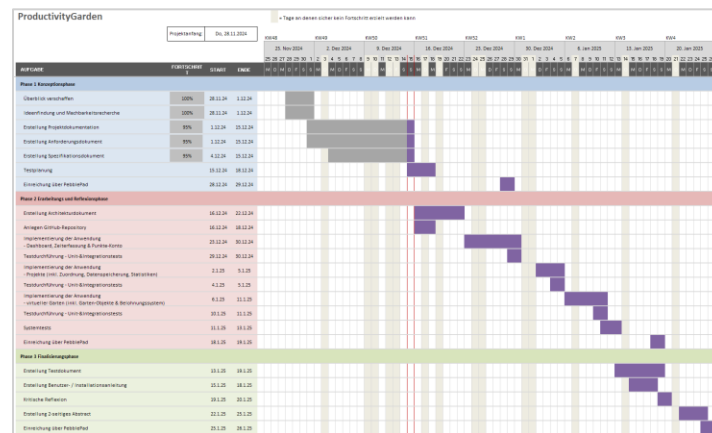
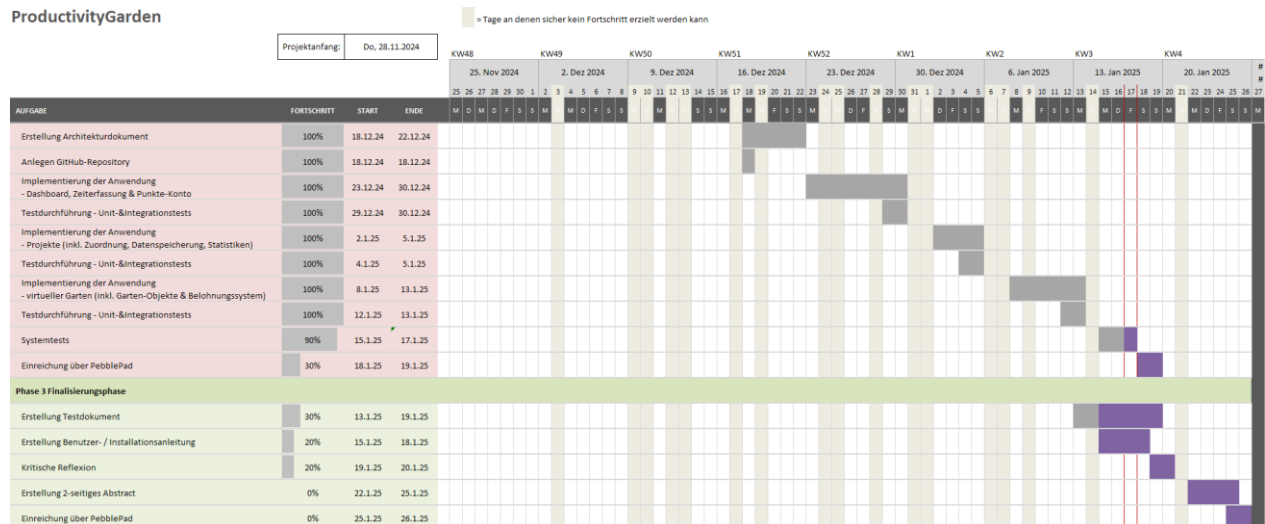


Abbildung 1 - Gantt-Diagramm

Stand 17.01.2025:



5. Literaturverzeichnis

Huseynli, B., & Uslu, A. (2024). A Qualitative Study on the Definition and Concept of Gamification. *Journal of Economic Sciences: Theory & Practice*, 81(1), (S. 40–50).
<https://doi.org/10.61640/jestp.2024.81.01.03>

Matallaoui, A., Hanner, N., & Zarnekow, R. (2016). Introduction to Gamification: Foundation and Underlying Theories. In S. Stieglitz, C. Lattemann, S. Robra-Bissantz, R. Zarnekow, & T. Brockmann, *Gamification: Using Game Elements in Serious Contexts* (S. 3–18). Springer International Publishing AG.
<http://ebookcentral.proquest.com/lib/badhonnef/detail.action?docID=4710247>

Anforderungsdokument

Projektname: ProductivityGarden

Name

Jonas Huber

Matrikelnummer

IU14085128

Modul

Projekt: Software Engineering

DLMCSPSE01_D

Datum

17.01.2025

Inhalt

1. Zielgruppen	2
Kundennutzen	2
Abgrenzung zu ähnlichen Anwendungen	2
Primäre Zielgruppe	2
Sekundäre Zielgruppe	2
2. Funktionale Anforderungen	3
User-Stories	3
Verbesserung der Konzentration und Produktivität	3
Zeitmanagement	3
Motivation	4
Stressabbau und Kreativität	4
Abgrenzung zu ähnlichen Anwendungen	4
Use-Case-Diagramm	5
3. Nicht-funktionale Anforderungen	6
Leistung und Ausführung der Anwendung	6
Benutzerschnittstelle und Benutzerfreundlichkeit	6
Sicherheit	6
4. Glossar	7
5. Literaturverzeichnis	8

Abbildungsverzeichnis

Abbildung 1 - UML-Use-Case-Diagramm	5
---	---

1. Zielgruppen

Kundennutzen

- **Verbesserung der Konzentration:** Durch den Pomodoro-Timer, der gezielte Arbeitseinheiten und Pausen fördert.
- **Zeitmanagement:** Flexibel einstellbare Timer und Stoppuhren ermöglichen das Tracking von Aktivitäten.
- **Motivation:** Belohnungen im virtuellen Garten fördern die Motivation
- **Stressabbau und Kreativität:** Der virtuelle Garten sorgt für eine angenehme Atmosphäre.
- **Individuelle Anpassung:** Nutzer:innen können Projekte zuordnen, Zeit aufzeichnen und ihren Garten nach ihren Vorlieben gestalten.

Abgrenzung zu ähnlichen Anwendungen

- **Simple und effektiv:** Keine Formen von Anwendungen-Bindung und Gewinnerzielung wie Paywalls, Werbung oder tägliche Herausforderungen.
- **Gamification:** Fokus auf Belohnungen und Visualisierung statt auf strikte Prokrastination-Sperren oder Disziplinierung.
- **Breiter Anwendungsbereich:** Geeignet für Arbeit, Lernen, Hobbys oder Entspannung.

Primäre Zielgruppe

Die primäre Zielgruppe umfasst Studierende, Berufstätige und allgemein alle Personen, die ihre Produktivität steigern wollen und gleichzeitig eine spielerische Motivation schätzen mit entspannenden Gamification-Elementen. Junge Erwachsene und Studierende haben oft Herausforderungen im Zeitmanagement und Stressabbau und benötigen Motivation für Studium oder Arbeit, wofür sie gerne einfache und unterhaltsame Tools ohne Paywalls und Werbung nutzen. Berufstätige dagegen (insbesondere Wissensarbeiter) wollen oft ihre Produktivität steigern und Projektzeiten erfassen, ohne viel Aufwand und aufdringliche Anwendungen mit Werbung.

Beispiel-Zielgruppenprofil:

- Sophie, 22 Jahre – Studentin
- Lebensstil: Sophie jongliert Studium, Nebenjob und Freizeitaktivitäten.
- Verwendet die Anwendung, um Lernphasen produktiv zu gestalten und dabei ein wenig Stress wieder abzubauen.
- Sie schätzt die visuelle Belohnung und die Einfachheit.
- Hat wenig Geduld für Anwendungen mit Premium-Elementen und Werbung.

Sekundäre Zielgruppe

Die Anwendung kann auch für Eltern, Teilzeitbeschäftigte oder generell für alle Personen nützlich sein, die mit Hilfe der Zeiterfassung einen besseren Überblick über ihre Zeit gewinnen möchten. Diese können die flexiblen Timer und die Zeiterfassung für Projekte, Hobbys oder

Haushaltstätigkeiten nutzen. Die Motivation wird auch durch die spielerischen Gamification-Elemente gefördert und erfordert weder eine hohe Technikaffinität noch viel Aufwand.

Beispiel-Zielgruppenprofil:

- Julian, 40 Jahre – Hobby-Bäcker und Vater
- Lebensstil: Arbeitet halbtags und widmet seine Nachmittage mit seinem Hobby Backen und der Organisation des Haushalts
- Er verliert schnell den Überblick über die Zeit, die er in verschiedene Aufgaben investiert und verwendet deshalb die Anwendung, um einen Überblick über ihr Zeitmanagement zu bekommen
- Er schätzt den entspannenden Aspekt der Pflanzenvisualisierung und nutzt flexible Timer oder die Stoppuhr für sein Hobby oder Hausarbeiten.
- Schätzt ebenfalls, dass die Anwendung nicht überladen ist und flexibel genutzt werden kann.

2. Funktionale Anforderungen

Im Folgenden erfolgt eine detaillierte Auflistung der gewünschten Funktionen und Features der Anwendung. Die Anforderungen werden dabei zunächst textuell durch User Stories beschrieben und anschließend durch ein UML-Use-Case-Diagramm visualisiert.

User-Stories

Verbesserung der Konzentration und Produktivität

1. Als Nutzer:in möchte ich einen Pomodoro-Timer verwenden, damit ich meine Arbeitsphasen und Pausen strukturiert mit einem festen Rhythmus einteilen kann.
2. Als Nutzer:in möchte ich die Länge von Produktiv- und Pausenzeiten anpassen können, damit der Timer meinen individuellen Bedürfnissen entspricht.
3. Als Nutzer:in möchte ich einen individuell einstellbaren Timer für meine nächste Produktivphase verwenden, ohne feste Pausenzeit und Rhythmus.
4. Als Nutzer:in möchte ich die Anwendung ohne störende Benachrichtigungen nutzen können, damit ich fokussiert bleiben kann.
5. Als Nutzer:in möchte ich eine einfache und intuitive Navigation, damit ich ohne großen Aufwand alle Funktionen nutzen kann.

Zeitmanagement

6. Als Nutzer:in möchte ich eine Stoppuhr verwenden, um flexibel Aktivitäten zu tracken, ohne an feste Zeitintervalle gebunden zu sein.
7. Als Nutzer:in möchte ich Zeiten manuell eintragen können, falls ich vergessen habe, den Timer oder die Stoppuhr zu starten.
8. Als Nutzer:in möchte ich eine Übersicht über meine produktiv genutzte Zeit erhalten, damit ich meinen Fortschritt nachvollziehen kann.
9. Als Nutzer:in möchte ich meine aufgezeichnete Zeit spezifischen Projekten oder Kategorien zuordnen können, um meine Produktivität für unterschiedliche Aufgaben zu verfolgen.

10. Als Nutzer:in möchte ich eine Visualisierung über die aufgewendete Zeit für verschiedene Projekte oder Kategorien sehen, mitunter um diese miteinander vergleichen zu können.

Motivation

11. Als Nutzer:in möchte ich für meine aufgezeichnete produktive Zeit virtuelle Belohnungen erhalten, damit ich motiviert bleibe, kontinuierlich produktiv zu sein.
12. Als Nutzer:in möchte ich sehen, wie mein Garten wächst und sich entwickelt, basierend auf der Zeit, die ich produktiv genutzt habe.
13. Als Nutzer:in möchte ich meinen virtuellen Garten mit verschiedenen Pflanzen und Dekorationen gestalten, um ein visuelles Feedback für meine Fortschritte zu haben.

Stressabbau und Kreativität

14. Als Nutzer:in möchte ich eine beruhigende und minimalistische Benutzeroberfläche haben, damit die Anwendung mich nicht von meiner Arbeit ablenkt.
15. Als Nutzer:in möchte ich, dass mein virtueller Garten eine entspannte Atmosphäre bietet, um Stress während meiner Pausen abzubauen.
16. Als Nutzer:in möchte ich in meiner Pausenzeit nicht tiefer in komplexe Spielmechaniken eintauchen und mich mit verschiedenen Inhalten auseinandersetzen müssen, sondern die Anwendung ohne viel Aufwand verwenden.
17. Als Nutzer:in möchte ich die Ästhetik meines Gartens individuell anpassen können, um meine Kreativität auszuleben.

Abgrenzung zu ähnlichen Anwendungen

18. Als Nutzer:in möchte ich auf Paywalls verzichten können, um die Anwendung ohne Einschränkungen zu nutzen.
19. Als Nutzer:in möchte ich keine täglichen Aufgaben oder Challenges vorgegeben bekommen, um mich nicht unter Druck gesetzt zu fühlen.
20. Als Nutzer:in möchte ich Belohnungen basierend auf meiner echten Produktivität erhalten, anstatt durch kostenpflichtige Optionen bevorzugt zu werden.
21. Als Nutzer:in möchte ich keine visuellen Reize haben und nicht durch Werbung und überladene Inhalte abgelenkt werden.

Use-Case-Diagramm

Das Use-Case-Diagramm (siehe Abbildung 1) veranschaulicht die Hauptinteraktionen der Nutzer:innen mit der Anwendung "ProductivityGarden". Es umfasst die verschiedenen Bereiche mit Zeiterfassung (z. B. Pomodoro-Timer), Zeitmanagement (z. B. Zuordnung von Produktivzeiten zu Projekten) und den virtuellen Gärten (z. B. Pflanzen und Dekorationen kaufen). Die Diagrammstruktur zeigt die einfache und zielgerichtete Benutzerführung der Anwendung.

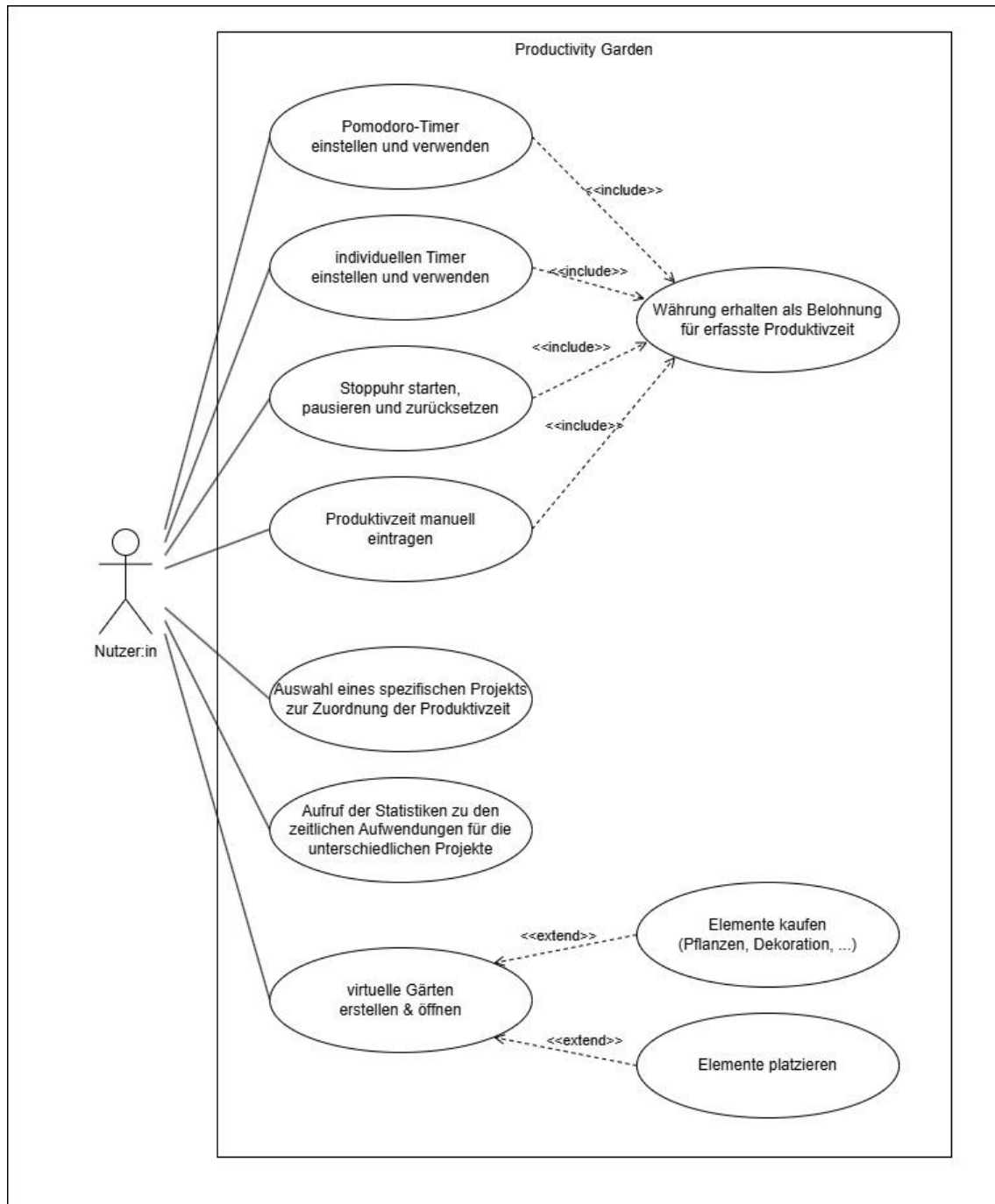


Abbildung 1 - UML-Use-Case-Diagramm

3. Nicht-funktionale Anforderungen

In diesem Kapitel werden die maßgeblichen Qualitätsanforderungen dargelegt, wie beispielsweise Leistung, Sicherheit und Benutzerfreundlichkeit.

Leistung und Ausführung der Anwendung

Die Anwendung soll so konzipiert werden, dass sie schnell und reaktionsfreudig arbeitet, was besonders wichtig ist, da es sich im Kern um ein Spiel handelt. Die Anwendung soll auf dem Betriebssystem Windows lauffähig sein und ohne zusätzliche Installationsschritte direkt verwendet werden können. Um den Nutzerkomfort zu erhöhen, bleibt die Anwendung auch im Hintergrund aktiv, sodass der Spielverlauf nicht unterbrochen wird, wenn der Nutzer zu einer anderen App wechselt. Darüber hinaus speichert die Anwendung relevante Daten, sodass der aktuelle Stand beim erneuten Starten nahtlos fortgesetzt werden kann. Diese Eigenschaften gewährleisten eine konsistente und angenehme Benutzererfahrung.

Benutzerschnittstelle und Benutzerfreundlichkeit

Bei der Gestaltung der Benutzerschnittstelle liegt der Fokus auf intuitiver und einfacher Bedienung, sodass Nutzer schnell und problemlos auf alle Funktionen zugreifen können. Die Oberfläche soll bewusst minimalistisch gehalten werden, um Ablenkungen und unnötige visuelle Reize zu vermeiden, sodass sich die Nutzer ganz auf das Spielerlebnis konzentrieren können. Darüber hinaus verzichtet die Anwendung vollständig auf Premium-Elemente, Paywalls, Werbung oder Wartezeiten, um eine uneingeschränkt zugängliche und störungsfreie Nutzung zu ermöglichen.

Sicherheit

Für die Sicherheit und den Schutz der Nutzerdaten werden alle relevanten Datenschutzbestimmungen eingehalten, einschließlich der Richtlinien der Datenschutz-Grundverordnung (DSGVO). Dies gewährleistet, dass personenbezogene Daten sicher verarbeitet und gespeichert werden. Es ist zu gewährleisten, dass keinerlei Daten an einen Server übermittelt werden müssen, sondern ausschließlich auf der ausführenden Instanz in privater Form gespeichert werden.

4. Glossar

Gamification	auch Spielifizierung genannt, bezeichnet die Anwendung spielerischer Elemente und Mechanismen in einem nicht spielerischen Kontext, z.B. im Bildungsbereich (Huseynli, 2024, S. 45).
Paywall	Eine Paywall ist eine digitale Schranke, die Inhalte in Anwendungen nur gegen Bezahlung zugänglich macht, z.B. durch ein Abonnement oder eine einmalige Zahlung (Ryte, 2021).
Pomodoro-Timer	Es handelt sich um eine Methode des Zeitmanagements. Es wird abwechselnd 25 Minuten konzentriert gearbeitet und 5 Minuten Pause gemacht (Pairan, 2024).

5. Literaturverzeichnis

- Huseynli, B., & Uslu, A. (2024). A Qualitative Study on the Definition and Concept of Gamification. *Journal of Economic Sciences: Theory & Practice*, 81(1), (S. 40–50).
<https://doi.org/10.61640/jestp.2024.81.01.03>
- Pairan, M. (2024, Oktober 28). Pomodoro-Technik: Mehr schaffen in 25 Minuten? *Gesunde Produktivität - Der Blog*. <https://www.marapairan.de/pomodoro-technik/>
- Ryte. (2021). Was ist eine Paywall? *Ryte Wiki*. <https://de.ryte.com/wiki/Paywall/>

Spezifikationsdokument

Projektname: ProductivityGarden

Name

Jonas Huber

Matrikelnummer

IU14085128

Modul

Projekt: Software Engineering

DLMCSPSE01_D

Datum

17.01.2025

Inhalt

1. Datenmodell	2
Textuelle Beschreibung	2
Visualisierung durch UML-Klassendiagramm	3
2. Geschäftsprozesse	4
Textuelle Beschreibung	4
Visualisierung durch UML-Aktivitätsdiagramm	5
3. Geschäftsregeln	6
Geschäftsregeln für Geschäftsobjekte	6
Geschäftsregeln für Geschäftsprozesse	7
4. Systemschnittstellen	8
Datenformat und Speicherstruktur	8
Systemintegration	8
5. Benutzerschnittstelle	9
Struktur und Inhalt der wichtigsten Dialoge	9
Hauptfenster	9
Virtueller Garten	11
Einstellungen	12
Dialogflüsse	12
Hauptfenster	12
Übersicht virtuelle Gärten	13
Virtueller Garten	13
Eingabevalidierung	13

Abbildungsverzeichnis

Abbildung 1 – Datenmodell UML-Klassendiagramm	3
Abbildung 2 – UML-Aktivitätsdiagramm zu Prozess „Pomodoro-Timer“	5
Abbildung 3 – Mockup Hauptfenster	10
Abbildung 4 – KI-generiertes Mockup Nr.1	10
Abbildung 5 – KI-generiertes Mockup Nr.2	10
Abbildung 6 – Mockup Übersicht über die virtuellen Gärten	11
Abbildung 7 – Mockup virtueller Garten	12

1. Datenmodell

Dieses Kapitel beinhaltet eine textuelle Beschreibung der zentralen Geschäftsobjekte der Anwendung und deren Beziehungen zueinander sowie einer Visualisierung in Form eines UML-Klassendiagramms.

Textuelle Beschreibung

1. Nutzer:innen

Nutzer:innen interagieren mit der Anwendung: Sie erstellen Projekte, messen Produktivzeiten, sammeln Punkte, erstellen Gärten und kaufen Elemente (Pflanzen und Dekoration).

2. Projekt

Ein Projekt steht für eine Aufgabe oder Aktivität, der die Nutzer:innen Zeit widmen.

3. Sitzung

Eine Sitzung repräsentiert eine Zeitspanne, die die Nutzer:innen für ein bestimmtes Projekt gearbeitet hat und somit dafür eine gewisse Anzahl an Punkten verdient.

4. Punkte-Konto

Das Punkte-Konto speichert die erforderlichen Daten für das Punktesystem. Es kann nur ein Punkte-Konto geben und dieses ist explizit zu den Nutzer:innen zugewiesen.

5. Virtueller Garten

Ein virtueller Garten steht für eine Visualisierte Spielwelt, in der Inhalte, sogenannte Garten-Objekte, platziert werden. Nutzer:innen können mehrere virtuelle Gärten erstellen.

6. Garten-Objekt

Ein Garten-Objekt repräsentiert ein Element, beispielsweise eine Pflanze oder Dekoration, das im Gegenzug zu Punkten in einem virtuellen Garten platziert wird.

Visualisierung durch UML-Klassendiagramm

Die Visualisierung in Abbildung 1 zeigt die zentralen Geschäftsobjekte der Anwendung, ihre Attribute und die Beziehungen untereinander.

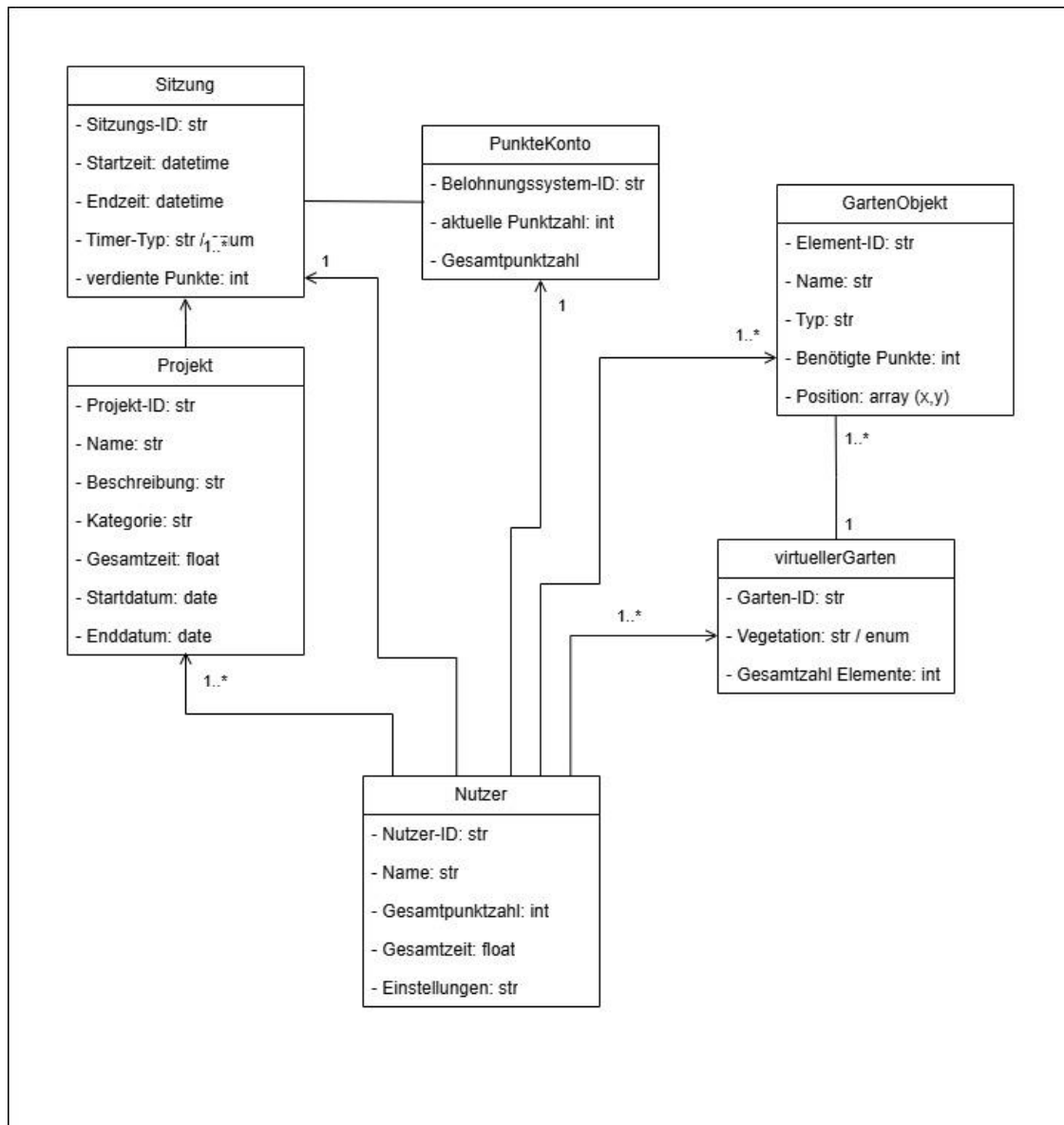


Abbildung 1 – Datenmodell UML-Klassendiagramm

Notiz (#1-01/25):

„Nutzer“ Klasse wurde in der Architekturentwicklung wieder rausgeworfen und für irrelevant empfunden, da nur eine Person die Anwendung verwendet und es keine Profil-/Nutzerverwaltung gibt.

2. Geschäftsprozesse

Im vorliegenden Kapitel erfolgt eine Darlegung der zentralen Funktionen und Prozesse der Anwendung. Zunächst erfolgt eine textuelle Darlegung und anschließend wird ein ausgewählter Kernprozess anhand eines UML-Aktivitätsdiagramms visualisiert.

Textuelle Beschreibung

1. Timer-Funktionalitäten

Nutzer:innen können zwischen verschiedenen Zeiterfassungsmethoden wählen:

- Pomodoro-Timer: Die Nutzer:innen arbeiten in Intervallen, z. B. 25 Minuten Arbeit und 5 Minuten Pause.
- Flexibler Timer: Die Nutzer:innen legen individuelle Zeitspannen fest.
- Stoppuhr: Startet und stoppt die Zeit flexibel, ohne festen Rahmen.

Nach Ablauf des Timers oder beenden der Stoppuhr erhalten die Nutzer:innen Punkte

Notiz (#2-01/25):

Nutzer:innen bekommen bereits während des Laufens der Timer / Stoppuhr Punkte, hauptsächlich aus Persistenz-Gründen. Die Nutzer:innen könnten früher als ursprünglich geplant fertig sein und falls die Anwendung aus Versehen geschlossen wird (zum Beispiel auch beim Wechsel auf zum „VirtualGarden“), wurden trotzdem die Punkte bereits gutgeschrieben.

2. Punktesystem

Punkte werden basierend auf produktiv verbrachter Zeit gesammelt (z.B. 1 Punkt pro 10 Minuten). Punkte können verwendet werden, um Objekte für einen virtuellen Garten zu kaufen

3. Projekt-Tracking

Nutzer:innen können ihre Zeit einem bestimmten Projekt zuweisen. Produktivzeiten werden für die jeweiligen Projekte in einer Statistik erfasst und visualisiert.

4. Virtuelle Gärten

Nutzer:innen haben die Möglichkeit, virtuelle Gärten zu erstellen. Nach der Erstellung umfasst ein solcher Garten lediglich ein leeres Raster, auf dem Objekte hinzugefügt werden können.

5. Datenspeicherung und Datenschutz

Alle Daten werden lokal auf dem Gerät der Nutzer:innen gespeichert und es erfolgt keine Übertragung an externe Server. Nutzerdaten wie Punkte, virtuelle Gärten und deren Objekte und Timer-Einstellungen werden bei jedem Start der App wiederhergestellt. Diese Daten werden während der Ausführung der Anwendung automatisch gespeichert.

6. Hintergrundnutzung

Die Anwendung läuft im Hintergrund weiter, sodass andere Anwendungen parallel genutzt werden können.

Visualisierung durch UML-Aktivitätsdiagramm

Zur Veranschaulichung wird einer der Kernprozesse mittels eines UML-Aktivitätsdiagramms visualisiert. In Abbildung 2 ist ein Teil der Timer-Funktionalitäten dargestellt, wobei es sich um die Pomodoro-Timer-Funktionalität handelt.

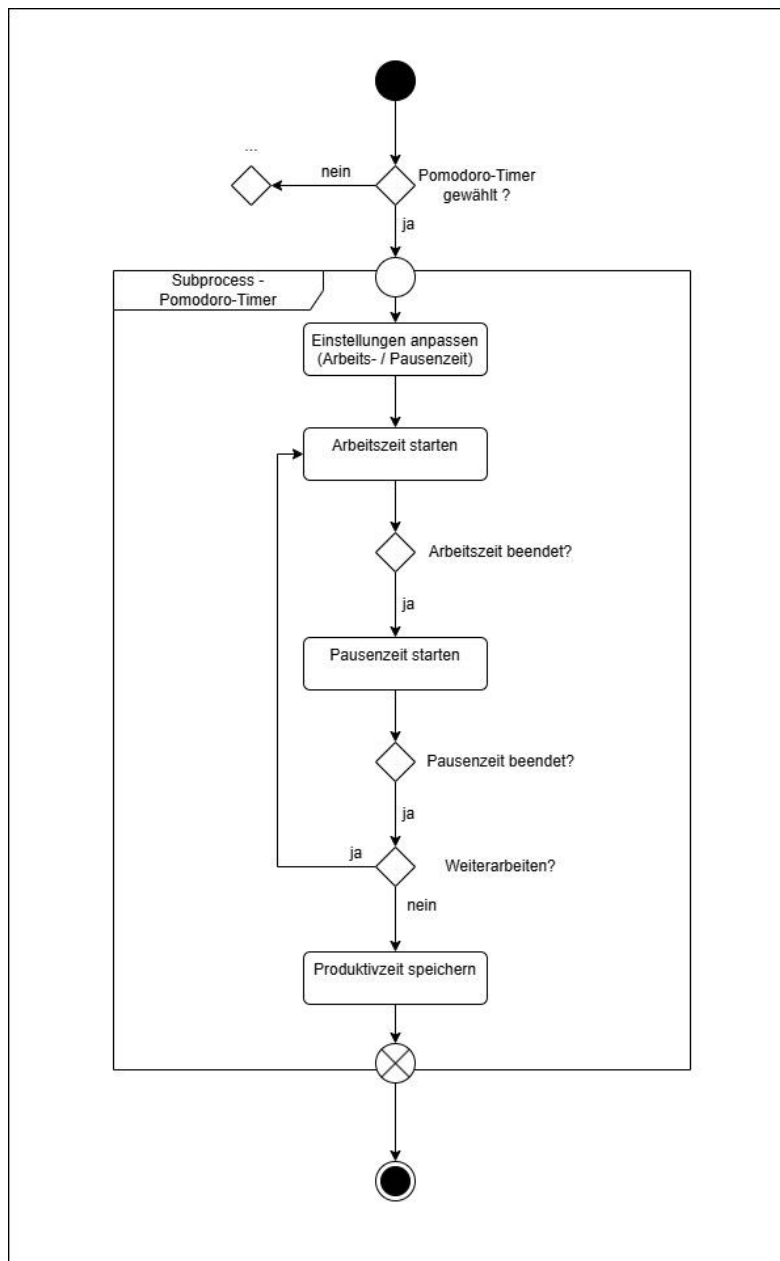


Abbildung 2 – UML-Aktivitätsdiagramm zu Prozess „Pomodoro-Timer“

3. Geschäftsregeln

In folgendem Abschnitt werden die zentralen Geschäftsregeln für die Anwendung ProductivityGarden definiert. Diese Regeln dienen als Grundlage für die Gestaltung der Geschäftsobjekte und -prozesse und stellen sicher, dass die Funktionen der Anwendung sowohl den Nutzeranforderungen als auch den Projektzielen entsprechen. Durch die Festlegung dieser verbindlichen Richtlinien wird ein reibungsloser Ablauf sowie eine intuitive und zuverlässige Nutzung der Anwendung gewährleistet.

Geschäftsregeln für Geschäftsobjekte

1. Zeiterfassung

- Der Timer muss eine Mindestzeit von 1 Minute und eine Höchstzeit von 24 Stunden unterstützen.
- Beim Pomodoro-Timer sind standardmäßig 25 Minuten Arbeit und 5 Minuten Pause eingestellt, die Zeiten können jedoch von Nutzer:innen individuell angepasst werden.
- Die Timer müssen jederzeit pausiert und beendet werden können
- Die Stoppuhr muss es ermöglichen, dass deren Betrieb jederzeit pausiert, wieder aufgenommen und beendet werden kann.

2. Punktesystem

- Nutzer:innen erhalten 1 Punkt für jede 10 Minuten produktiver Zeit
- Punkte können ausschließlich innerhalb der Anwendung für den Kauf von virtuellen Objekten eingesetzt werden
- Es ist nicht möglich, Punkte in der Anwendung zu „verlieren“ oder abgezogen zu bekommen

3. Projekt-Tracking

- Jede produktiv verbrachte Zeit muss einem Projekt oder einer Aufgabe zugeordnet werden
- Projekte können gelöscht oder archiviert werden, wenn dies von Nutzer:innen gewünscht ist
Notiz (#3-01/25):
Projekte können nicht archiviert werden. Feature aus zeitlichen Gründen nicht umgesetzt.
- Das Enddatum des Projekts darf leer gelassen werden, darf jedoch niemals vor dessen Startdatum liegen

4. Virtueller Garten

- Nutzer:innen können beliebig viele Gärten erstellen
- Ein virtueller Garten muss ein Raster enthalten, auf dem Objekte platziert werden können
- Ein Objekt darf nur in einem Garten platziert werden, wenn der Nutzer:innen über genügend Punkte verfügt

- Ein Objekt kann nur platziert werden, wenn es in das gewählte Raster passt und dieses nicht blockiert ist durch ein bereits platziertes Objekt

Notiz (#4-01/25):

Aus zeitlichen Gründen konnte nicht das Bewegen von bereits platzierten Objekten implementiert werden. Wenn nun ein Objekt also platziert wurde, würde es laut dieser Regel für immer da platziert sein. Da aber eventuell ein anderes Objekt platziert werden möchte an genau dieser Stelle, wird das bisherige Objekt aktuell noch überschrieben.

Geschäftsregeln für Geschäftsprozesse

1. Zeiterfassungsprozesse

- Eine laufende Zeiterfassung darf nicht verändert werden. Änderungen können erst nach der Beendigung oder dem Abbrechen der Zeiterfassung vorgenommen werden.
- Pomodoro-Timer: Nach Ablauf der eingestellten Zeit müssen die Nutzer:innen eine Aktion wählen (z.B. „neuen Zyklus starten“), bevor ein neuer Zyklus beginnt.

2. Punktevergebeprozess

- Punkte werden nur bei abgeschlossenen Zeiterfassungen gutgeschrieben. Dazu zählen Timer und Stoppuhren deren Zeit abgelaufen ist oder die explizit „beendet“ wurden.

Siehe Notiz #2-01/25

- Beim Kauf eines Objekts, werden die entsprechenden Punkte sofort vom Kontostand abgezogen

3. Datenmanagementprozess

- Alle Nutzerdaten (z.B. Punkte, Timer-Einstellungen, virtuelle Gärten) müssen lokal und automatisch gespeichert werden.
- Beim Start der Anwendung müssen die zuletzt gespeicherten Daten geladen werden
- Es ist sicherzustellen, dass keine personenbezogenen Daten an Dritte weitergegeben werden oder auf externen Servern gespeichert werden

4. Nutzerinteraktionsprozesse

- Die Benutzeroberfläche muss sicherstellen, dass Benutzeraktionen wie das Löschen von Gärten oder Projekten oder sonstigen Daten eine Bestätigung erfordern, um versehentliche Löschungen zu vermeiden.
- Jede Aktion, die eine irreversible Auswirkung hat (z. B. Kauf von Objekten), muss den Nutzer:innen vorher klar kommuniziert werden.

4. Systemschnittstellen

In diesem Kapitel werden die technischen Schnittstellen der Anwendung beschrieben. Dazu gehören die genutzten Protokolle, Datenformate und Mechanismen, die für die lokale Speicherung sowie die Integration mit dem Betriebssystem Windows erforderlich sind.

Datenformat und Speicherstruktur

JSON eignet sich ideal für die flexible Speicherung strukturierter Daten wie Timer-Einstellungen und Punktestände. Für eine komplexere Datenverwaltung, beispielsweise die Verknüpfung von Projekten mit Zeitdaten, bietet sich SQLite als effiziente relationale Datenbanklösung an.

Systemintegration

Um Benachrichtigungen und Interaktionen innerhalb des Systems zu implementieren, wie z.B. Toast-Benachrichtigungen bei Ablauf des Timers, können Systemaufrufe an die Windows-API gesendet werden. Python-Bibliotheken wie „win10toast“ oder „plyer“ bieten hierfür praktische Lösungen.

Notiz #5-01/25:

Da das Fehlen dieses Features nicht die „Funktionalität“ beeinträchtigt, wurde es aus zeitlichen Gründen nicht umgesetzt.

5. Benutzerschnittstelle

Um eine Benutzeroberfläche für die Anwendung zu definieren, werden die Anforderungen in drei Kategorien gegliedert:

Struktur und Inhalt der wichtigsten Dialoge

Die Anwendung besteht aus verschiedenen Fenstern (Dialogen), deren Struktur und Inhalt definiert werden.

Hauptfenster

Inhalt:

- Timer-Bereich:
 - Pomodoro-Timer mit voreingestellten 25-5 Minuten, anpassbar
 - Timer und Stoppuhr, individuell konfigurierbar
- Punkteanzeige: Anzahl gesammelter Punkte insgesamt und verfügbare Punkte
- Projekte-Bereich:
 - Auswahl eines Projekts über ein Drop-Down-Menü
 - Erstellung eines Projekts
 - Übersicht über relevante Infos des aktuell ausgewählten Projekts wie gesamt verbrachte Produktivzeit, Startdatum und Enddatum (optional)
- Notizen-Bereich:
 - Ein Bereich mit einem Textfeld in den Notizen und Gedanken geschrieben werden können.
- Navigation: Tabs oder Buttons zu „Virtuelle Gärten“, „Einstellungen“ und „Statistiken“

Aufbau:

- Minimalistisch mit klaren, ablenkungsfreien Designs
- Aufteilung in verschiedene Bereiche:
 - Punkteanzeige
 - Timer-Bereich
 - Projekte-Bereich
 - Notizen-Bereich
- Anzeige von Fortschrittbalken (z.B. für die verbleibende Zeit im Timer)

Mockups:

Zur Visualisierung und damit zum besseren Verständnis der oben genannten Punkte wurde ein Mockup erstellt, das in Abbildung 3 dargestellt ist. Dieses ist ein reiner Prototyp, der nur den Ansatz des Aussehens des Hauptfensters zeigen soll. Die beiden KI-generierten Bilder in Abbildung 4 und Abbildung 5 können als weitere Inspiration z.B. für die Gestaltung der Oberfläche oder bestimmter Elemente dienen.

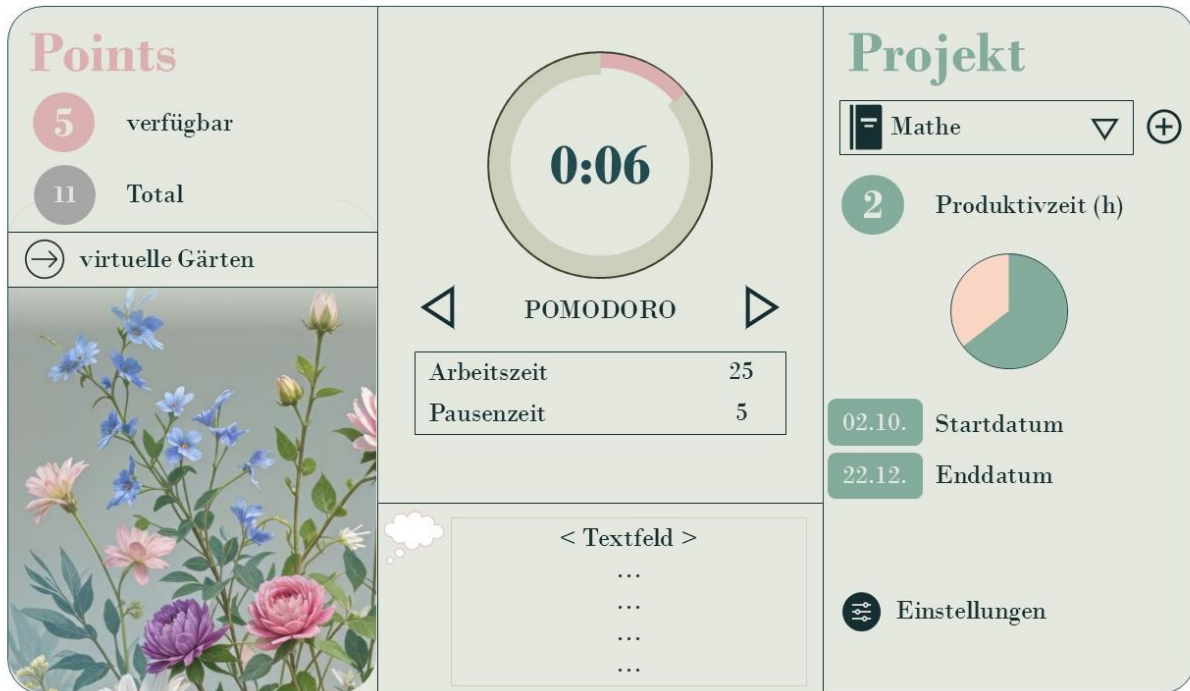


Abbildung 3 – Mockup Hauptfenster



Abbildung 4 – KI-generiertes Mockup Nr.1



Abbildung 5 – KI-generiertes Mockup Nr.2

Virtueller Garten

Inhalt:

- Übersicht der angelegten Gärten
- Ansicht eines expliziten Gartens
 - Raster, auf das Objekte (Pflanzen und Dekorationen) hinzugefügt werden können
 - Menü mit Objektauswahl

Aufbau:

- Übersicht der Gärten zeigt die Vorschau eines Gartens im Vordergrund und eine Schaltfläche zum Durchklicken der Gärten
- Button zum Anlegen eines neuen Gartens
 - Aufforderung zur Bearbeitung der Informationen des neuen Gartens:
 - Name
 - Vegetation
- Nach Auswahl eines Gartens verschwindet die Übersicht der Gärten
 - Fokus auf den ausgewählten Garten
 - Buttons, um wieder zur Übersicht der Gärten oder zum Dashboard zu kommen
- Auswahlbereich mit verschiedenen Objekten zur Platzierung unten (nur wenn Fokus auf einen expliziten Garten)
- Punkteanzeige

Mockups:

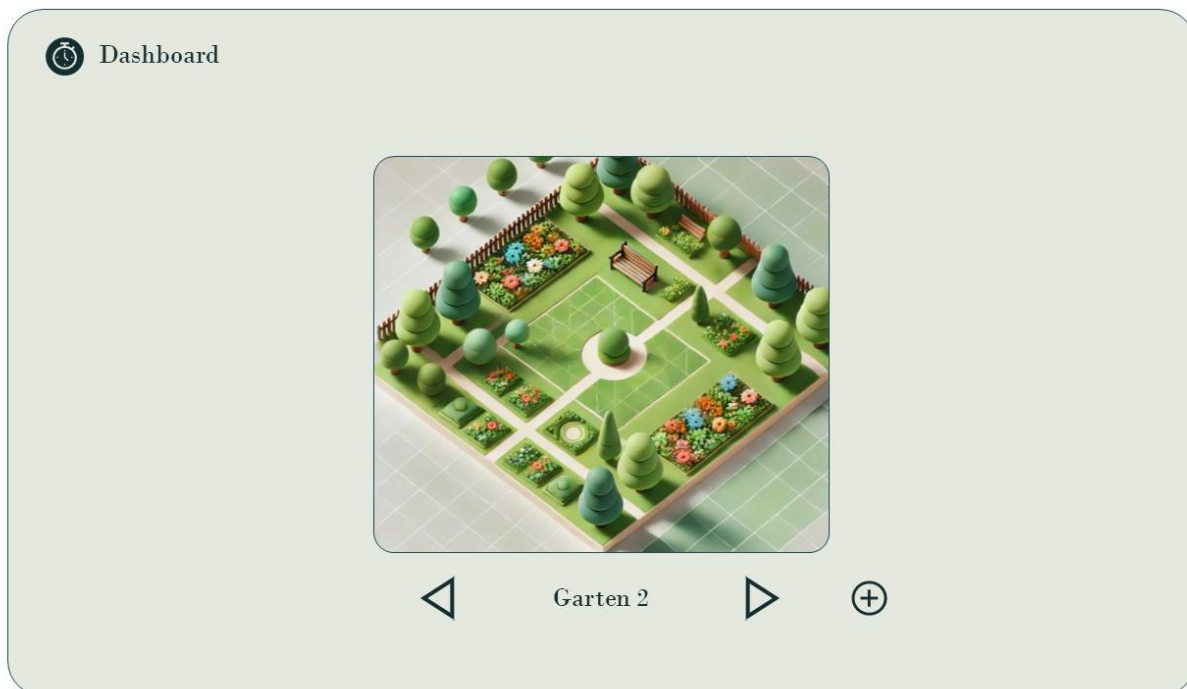


Abbildung 6 – Mockup Übersicht über die virtuellen Gärten

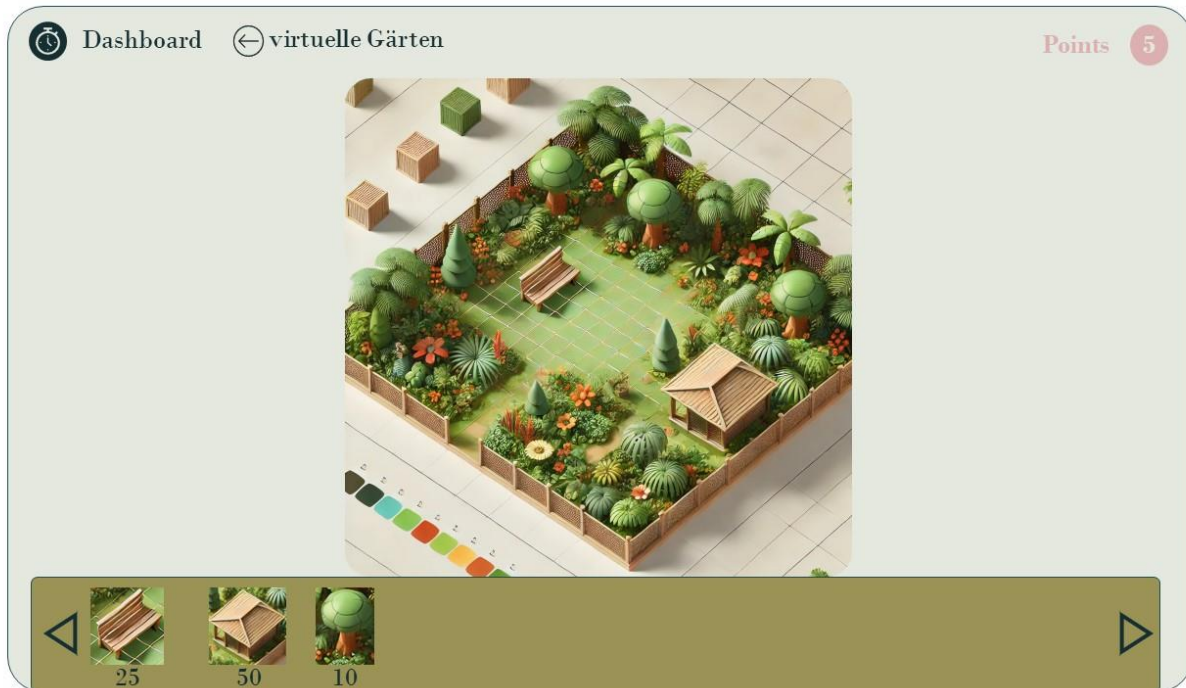


Abbildung 7 – Mockup virtueller Garten

Einstellungen

Inhalt:

- Möglichkeiten zum Löschen von Daten:
 - Alle Daten löschen
 - Explizite Projekte löschen
 - Explizite virtuelle Gärten löschen

Dialogflüsse

Dialogflüsse beschreiben die Abfolge von Interaktionen zwischen den Nutzer:innen und der Anwendung, insbesondere darauf, wie verschiedene Dialoge (z. B. Fenster, Pop-ups oder Bildschirme) miteinander verbunden sind und wie Nutzer:innen durch diese Dialoge navigieren können.

Hauptfenster

(Startbildschirm)

Timer-Bereich

- Klick auf Uhr -> Starten eines Timers oder Stoppuhr
- Klick auf Uhr während läuft -> Pausieren des laufenden Timers / der Stoppuhr
- Klick auf Pfeilsymbole unter der Uhr -> Auswahl eines Timers oder Stoppuhr
- Klick auf Zahlen „Arbeitszeit“, „Pausenzeit“, ... -> Bearbeitung der Laufzeit der Timer

Projekt-Bereich

- Klick auf Projektmenü -> Auswahl eines Projekts
- Klick auf Plussymbol neben dem Projektmenü -> Erstellung eines Projekts

Textfeld

- Klick auf Textfeld im unteren Bereich -> Möglichkeit zur Bearbeitung des Texts

Navigationsbuttons

- Klick auf „Einstellungen“ - Schriftzug -> Öffnen des Fensters „Einstellungen“
- Klick auf „virtuelle Gärten“ – Schriftzug -> Navigieren zum Fenster „Übersicht virtuelle Gärten“

Übersicht virtuelle Gärten

- Klick auf aktuell angezeigten Garten -> Auswahl des Gartens
- Klick auf die Pfeilsymbole -> Durchstöbern der angelegten Gärten
- Klick auf Plussymbol -> Anlegen eines neuen Gartens
- Klick auf „Dashboard“ – Schriftzug -> Navigieren zum Fenster „Hauptfenster“

Virtueller Garten

- Auswahl eines platzierten Objekts im virtuellen Garten -> Verschieben oder Entfernen des Objekts
- Auswahl eines Objekts aus dem Kauf-Bereich -> Kauf und Platzierung des Objekts im virtuellen Garten
- Klick auf „Dashboard“ – Schriftzug -> Navigieren zum Fenster „Hauptfenster“
- Klick auf „virtuelle Gärten“ – Schriftzug -> Navigieren zum Fenster „Übersicht virtuelle Gärten“

Eingabevalidierung

Die Eingabevalidierungen stellen sicher, dass alle Eingaben korrekt und sinnvoll sind.

Timer-Einstellungen

- Zeitwerte müssen positiv sein
- Zeitwerte dürfen eine festgelegte Grenze nicht überschreiten
 - Beim flexibel anpassbaren Timer maximal 24 Stunden bzw. 1440 Minuten
 - Beim Pomodoro-Timer darf die Arbeitszeit maximal 300 Minuten und die Pausenzeit maximal 120 Minuten betragen

Projekte

- Der Projektname darf nicht leer sein
- Der Projektname darf maximal 40 Zeichen lang sein
- Sonderzeichen sind erlaubt
- Startdatum muss ein Datum sein
 - Format überprüfen -> Eingabe von 1.2. bis zu 01.02.2020 ist erlaubt
 - Regulärer Ausdruck: `^(0?[1-9]]{12}[0-9]]3[01])\.(0?[1-9]]1[0-2])\.\(d{4})?$`

- Enddatum darf leer gelassen werden
- Enddatum muss ein Datum sein (falls nicht 0 bzw. leer)
- Enddatum muss im Vergleich zum Startdatum größer sein (in der Zukunft liegen)

Textfeld (auf dem Hauptfenster unter dem Timer-Bereich)

- Ausschließlich Text erlaubt
- Maximal 1000 Zeichen

Virtueller Garten

- Der Gartenname darf nicht leer sein
- Der Gartenname darf maximal 40 Zeichen lang sein
- Sicherstellen, dass die Platzierung eines Objekts nur in freien Rasterfeldern erfolgt,
[siehe Notiz #4-01/25](#)
- Kein Kauf von Objekten, wenn der Punktestand nicht ausreicht

Architekturdokument

Projektname: ProductivityGarden

Name

Jonas Huber

Matrikelnummer

IU14085128

Modul

Projekt: Software Engineering

DLMCSPSE01_D

Datum

15.01.2025

Inhalt

1. Technologieübersicht	2
Programmiersprache	2
Python	2
GUI-Framework	2
PyQt6.....	2
Zeitmanagement-Funktionalitäten	2
PyQt6.....	2
Spiel- und Grafikbibliothek	2
Pygame.....	2
Datenpersistenz	2
JSON.....	2
SQLite3.....	3
2. Architekturübersicht	4
Strang der Produktivanwendung	5
Präsentationsschicht	5
Logikschicht.....	5
Daten- und Persistenzschicht	5
Strang der Spielkomponente	5
Präsentations- und Logikschicht:.....	5
Daten- und Persistenzschicht:	6
3. Struktur	7
Hauptkomponenten der Anwendung	7
UML-Klassendiagramm.....	7
4. Verhalten des zentralen Systemablaufs: Pomodoro-Timer	9
Szenario:	9
Akteure:	9
UML-Sequenzdiagramm.....	9

1. Technologieübersicht

Die Auswahl der Technologien und Werkzeuge für die Entwicklung von ProductivityGarden wurde auf der Grundlage der funktionalen und nicht-funktionalen Anforderungen des Projekts getroffen. Die ausgewählten Technologien ermöglichen eine flexible, performante und benutzerfreundliche Umsetzung der Anforderungen von ProductivityGarden. Im Folgenden werden die verwendeten Technologien beschrieben und ihre Auswahl begründet.

Programmiersprache

Python

Python wurde aufgrund seiner vielfältigen Einsatzmöglichkeiten, einer großen Community und einer Vielzahl von Bibliotheken gewählt. Python ermöglicht eine schnelle Entwicklung und bietet hervorragende Unterstützung für plattformübergreifende Anwendungen.

GUI-Framework

PyQt6

Anwendung: Gestaltung der Benutzeroberfläche (GUI)

Begründung: PyQt6 bietet umfassende Funktionen zur Erstellung responsiver Desktop-GUIs. Es unterstützt eine Vielzahl von Widgets, die für die minimalistische und intuitive Gestaltung der Oberfläche von ProductivityGarden notwendig sind.

Zeitmanagement-Funktionalitäten

PyQt6

Anwendung: Implementierung der Timer-, Stoppuhr- und Pomodoro-Funktionen.

Begründung: Die Ereignissteuerung (Event Loop) von PyQt6 ermöglicht eine einfache Implementierung von zeitbasierten Funktionen wie Timer und Stoppuhren. Die Integration in das GUI-Framework gewährleistet eine reibungslose und performante Ausführung dieser Features.

Spiel- und Grafikbibliothek

Pygame

Anwendung: Rendering von grafischen Elementen, Animationen und Interaktionen in den virtuellen Gärten.

Begründung: Pygame ist eine leichtgewichtige Bibliothek, die sich ideal für Spielelemente eignet. Sie bietet Unterstützung für 2D-Grafiken und Benutzerinteraktionen, was für die Gestaltung der virtuellen Gärten essenziell ist.

Datenpersistenz

JSON

Anwendung: Speicherung von Benutzereinstellungen und Benutzerstatistiken.

Begründung: JSON ist ein leichtgewichtiger, plattformunabhängiger Standard für die Datenserialisierung. Es eignet sich ideal zur Speicherung von benutzerdefinierten Einstellungen und ermöglicht einen schnellen Lese- und Schreibzugriff.

SQLite3

Anwendung: Speicherung und Verwaltung komplexerer Datenstrukturen wie Projektinformationen.

Begründung: SQLite3 ist eine eingebettete Datenbank, die keine separate Serverinstallation erfordert. Sie ist robust, effizient und ermöglicht die Speicherung strukturierter Daten direkt auf dem Gerät des Nutzers, wodurch Datenschutzanforderungen (z. B. DSGVO) erfüllt werden.

2. Architekturübersicht

Die Architektur von "ProductivityGarden" basiert auf einer **Schichtenarchitektur** mit einer klaren Trennung zwischen Datenverwaltung, Logik und Benutzeroberfläche und bietet gleichzeitig Flexibilität für die unterschiedlichen Anforderungen der GUI (PyQt6) und des Spiels (Pygame).

Das System ist in zwei getrennte Stränge unterteilt, die jeweils aus mehreren Schichten bestehen. Die Trennung in PyQt6 und Pygame ermöglicht es, die Stärken beider Frameworks zu nutzen. PyQt6 sorgt für eine reaktionsschnelle und moderne GUI, während Pygame auf grafikintensive und interaktive Animationen spezialisiert ist.

Die Schichtenarchitektur bietet folgende Vorteile:

- Flexibilität: Die parallele Strangstruktur in der Präsentationsschicht ermöglicht eine optimale Nutzung von PyQt6 für GUI-Funktionen und Pygame für das grafikintensive Gamification-Element.
- Wartbarkeit und Erweiterbarkeit: Die Schichtenarchitektur mit klaren Abhängigkeiten erleichtert die Entwicklung und Erweiterung einzelner Komponenten.
- Leistung: Beide Stränge sind unabhängig voneinander und können so optimiert werden, um eine reaktionsfreudige und leistungsfähige Nutzung zu gewährleisten.
- Benutzerfreundlichkeit: Die Benutzerinteraktionen sind klar getrennt: die GUI für die Bedienung der Zeitfunktionen und das Projektmanagement und der Pygame-Strang für das Spielerlebnis.

Abbildung 1 zeigt die Schichtenarchitektur der Anwendung mit der bereits erwähnten Trennung zwischen der Produktivanwendung und der Spielkomponente. Beide Bereiche haben ihre eigenen Komponenten in jeder Schicht.

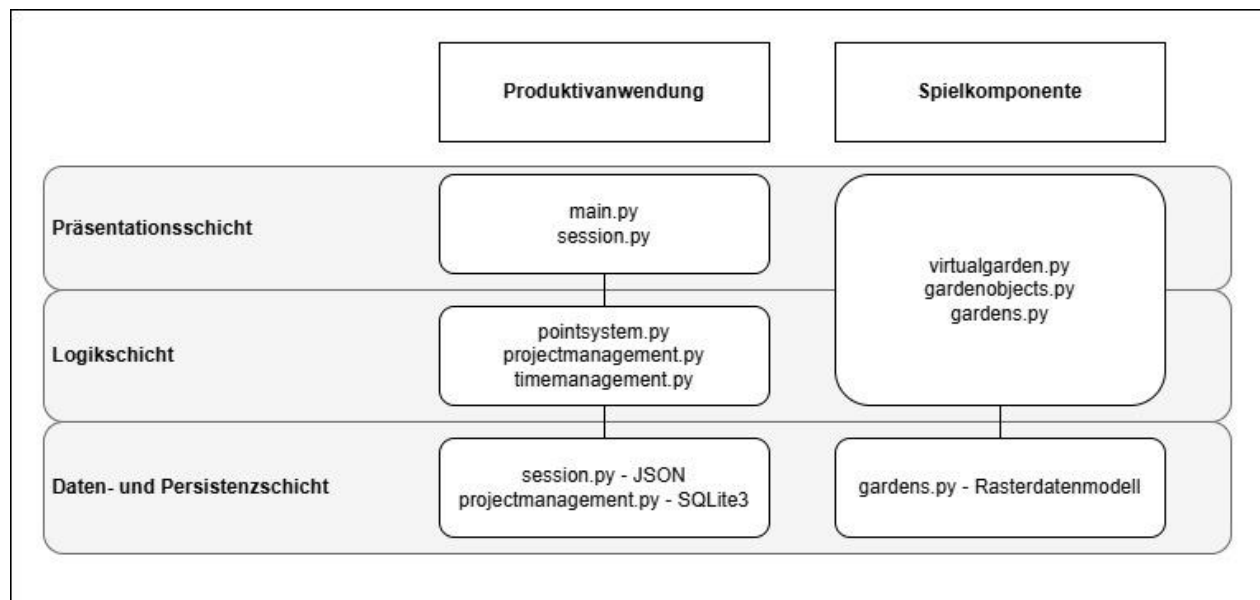


Abbildung 1 – Schichtenmodell der ProductivityGarden-Anwendung

Strang der Produktivanwendung

Präsentationsschicht

Diese Schicht enthält die Benutzeroberfläche, die mit dem PyQt6 Framework entwickelt wurde. Sie stellt das Hauptfenster der Anwendung dar, das für die Bedienung der Timer, die Projektverwaltung und die Konfiguration zuständig ist.

Komponenten:

- *main.py*: Startet die Anwendung und initialisiert die PyQt6-Oberfläche
- *session.py*: Beinhaltet die Logik und Darstellung der GUI

Logikschicht

Die Logikschicht enthält die zentrale Logik der Anwendung. Diese Schicht sorgt für:

- Verwaltung der Timer und Zeitmessungen
- Verwaltung der Projekte und Zuordnung von Zeit zu Projekten
- Berechnung und Verwaltung des Punkte-Systems

Die Logikschicht wurde modular und unabhängig von der Darstellung gestaltet, um Wiederverwendbarkeit und einfache Wartbarkeit sicherzustellen.

Komponenten:

- *pointssystem.py*: Punkteberechnung basierend auf produktiver Zeit
- *projectmanagement.py*: Verwaltung der Projekte und zugehörigen Arbeitszeiten
- *timemanagement.py*: Implementierung von Timer, Stoppuhr und Pomodoro-Funktion

Daten- und Persistenzschicht

Diese Schicht ist für die Speicherung und Verwaltung von Nutzer- und Projektdaten verantwortlich.

Komponenten:

- SQLite3: Speicherung persistenter Daten wie Projekte und Punkte
- JSON: Speicherung temporärer Daten und Konfigurationsdateien

Strang der Spielkomponente

Präsentations- und Logikschicht:

Umfasst die Darstellung des Gamification-Elements, inklusive virtuellem Garten mit Interaktionen und Animationen mittels Pygame. Da die Interaktionen, die hinterlegte Logik und die daraus resultierenden Reaktionen wie Animationen in einem Spiel verschmelzen, wurde hier die Präsentationsschicht und die Logikschicht zusammengeführt.

Komponenten:

- *virtualgarden.py*: Verantwortlich für das Rendering und die Interaktionen im virtuellen Garten
- *gardens.py*: Modellierung der virtuellen Gärten
- *gardenobjects.py*: Modellierung der Gartenobjekte

Daten- und Persistenzschicht:

Beinhaltet das Speichern und Verwalten von virtuellen Gärten und deren Spielstand, d.h. welche Objekte sind bereits an welcher Stelle platziert worden.

Komponenten:

- *gardens.py*: ermöglicht das Laden und Speichern eines virtuellen Gartens über ein Rasterdatenmodell

3. Struktur

In diesem Kapitel erfolgt die Beschreibung der Struktur der Anwendung. Der Fokus liegt hierbei auf den Hauptkomponenten und -klassen der Anwendung, deren Beziehungen und Abhängigkeiten zueinander sowie auf den wichtigsten Attributen und Methoden. Die visuelle Darstellung dieser Strukturen erfolgt mittels UML-Klassendiagramm.

Hauptkomponenten der Anwendung

Die Anwendung wird in die folgenden Hauptklassen unterteilt:

- `main()`: Dient als Startpunkt für die Ausführung der Anwendung
- `MainSession`: Hauptklasse für die GUI und den Produktivanwendungsteil
- `virtualgardens.py`: Hauptskript für die Spielkomponente mit den virtuellen Gärten
- `TimeManagement`: Verwaltung der Timer-Logik (Pomodoro, Stoppuhr, normaler Timer)
- `ProjectManagement`: Verwaltung der Projekte und produktiven Zeiten, sowie deren Speicherung.
- `PointsSystem`: Verwaltung der gesammelten und einsetzbaren Punkte.
- `Garden`: Verwaltung der virtuellen Gärten, inklusive Laden und Speichern der Daten.
- `GardenObject`: Repräsentiert Objekte, die in den virtuellen Gärten platziert werden können.

UML-Klassendiagramm

Abbildung 2 zeigt das UML-Klassendiagramm mit den Hauptkomponenten der Anwendung. Dieses dient der visuellen Darstellung der Struktur und der Beziehungen zwischen den Hauptkomponenten. Es bietet einen klaren Überblick über die Attribute und Methoden der einzelnen Klassen sowie deren Vererbungs- und Abhängigkeitsbeziehungen.

Hinweise

1. Die Gliederung in 3 Ebenen wurde bewusst gewählt, um eine Top-Down-Struktur widerzuspiegeln. Oben befindet sich der Startpunkt für die Ausführung der Anwendung mit der `main()` Funktion. Darunter befinden sich die beiden Hauptklassen für die beiden verschiedenen Teilkomponenten der Anwendung und darunter die verschiedenen Klassen, die zusätzliche Komponenten der Logik und des Datenmanagements übernehmen.
2. Die Methode `create_gui()` der Klasse `MainSession` enthält alle weiteren Komponenten der Klasse, die zur Erstellung der GUI implementiert sind. Da eine detaillierte Auflistung das Diagramm unübersichtlich machen würde und keinen zusätzlichen Nutzen bringt, wurden diese in einer Methode zusammengefasst.
3. Die beiden Hauptskripte „`virtualgardens.py`“ und „`session.py`“ (beinhaltet die `MainSession`-Klasse) der beiden geteilten Applikationen können sich gegenseitig aufrufen. Somit kann nahtlos von der einen Applikation zur anderen gewechselt werden.

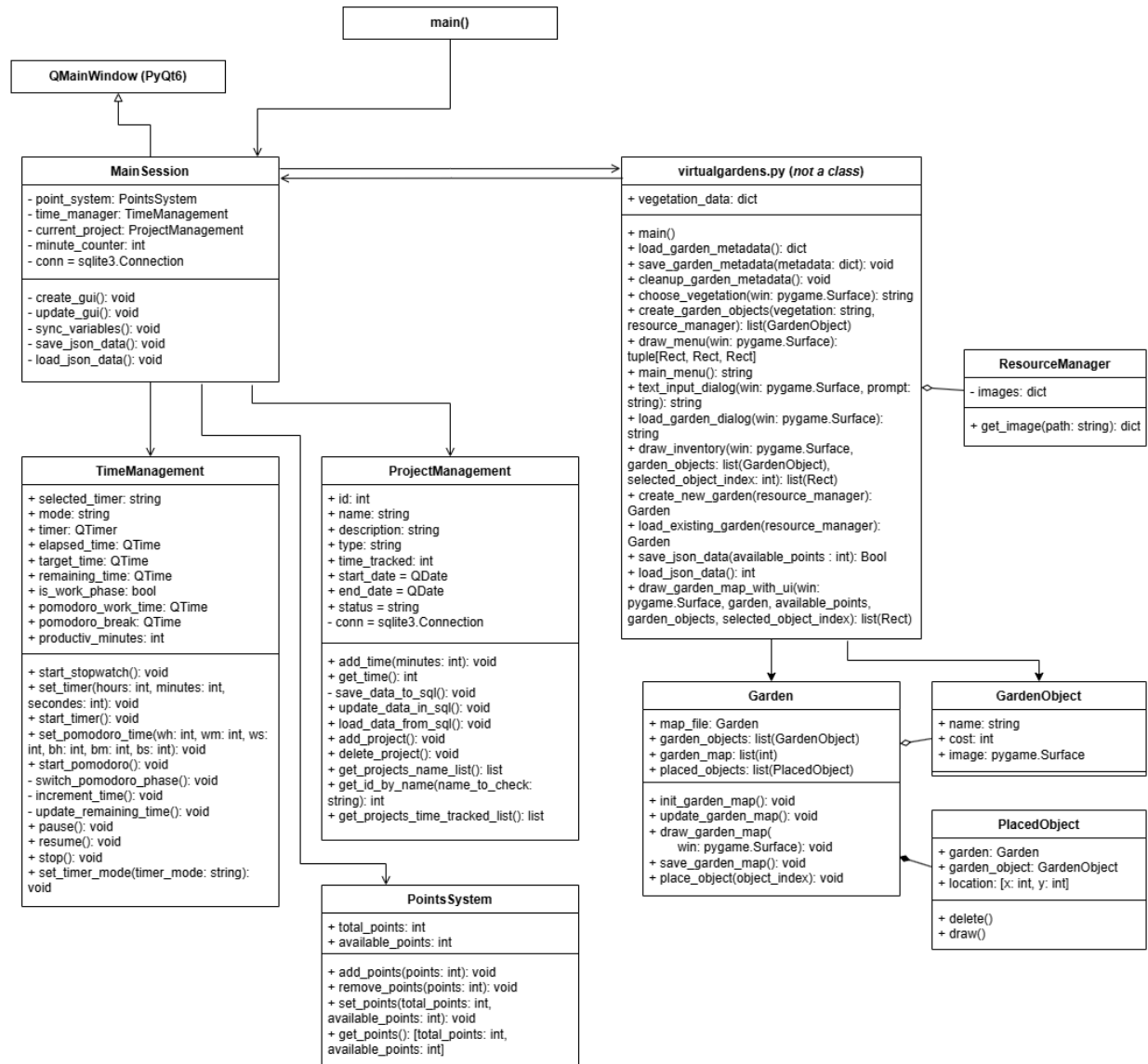


Abbildung 2 – UML-Klassendiagramm

4. Verhalten des zentralen Systemablaufs: Pomodoro-Timer

Im Folgenden wird ein zentraler Systemablauf eines Kernfeatures der Anwendung anhand des Systemablaufs des Pomodoro-Timers beispielhaft beschrieben und visualisiert.

Szenario:

Eine Person startet einen Pomodoro-Timer über die Benutzeroberfläche. Die Eingabe der Timer-Felder wird überprüft und wenn diese gültig sind, startet der Timer. Während der Timer läuft, erhöht sich die Produktivzeit des aktuell ausgewählten Projekts automatisch jede Minute und alle zehn Minuten wird ein Punkt auf dem Punktekonto gutgeschrieben.

Akteure:

Person: Interagiert mit der Benutzeroberfläche und startet den Timer

MainSession: Zuständig für die Benutzeroberfläche und Logik

TimeManagement: Verwalten des Timers

PointsSystem: Hinzufügen von Punkten

ProjectManagement: Speichern der Produktivzeit zum zugehörigen Projekt

UML-Sequenzdiagramm

Das in Abbildung 3 dargestellt UML-Sequenzdiagramm veranschaulicht den zentralen Systemablauf dar und visualisiert die Interaktionen zwischen den einzelnen Komponenten während eines typischen Ablaufes.

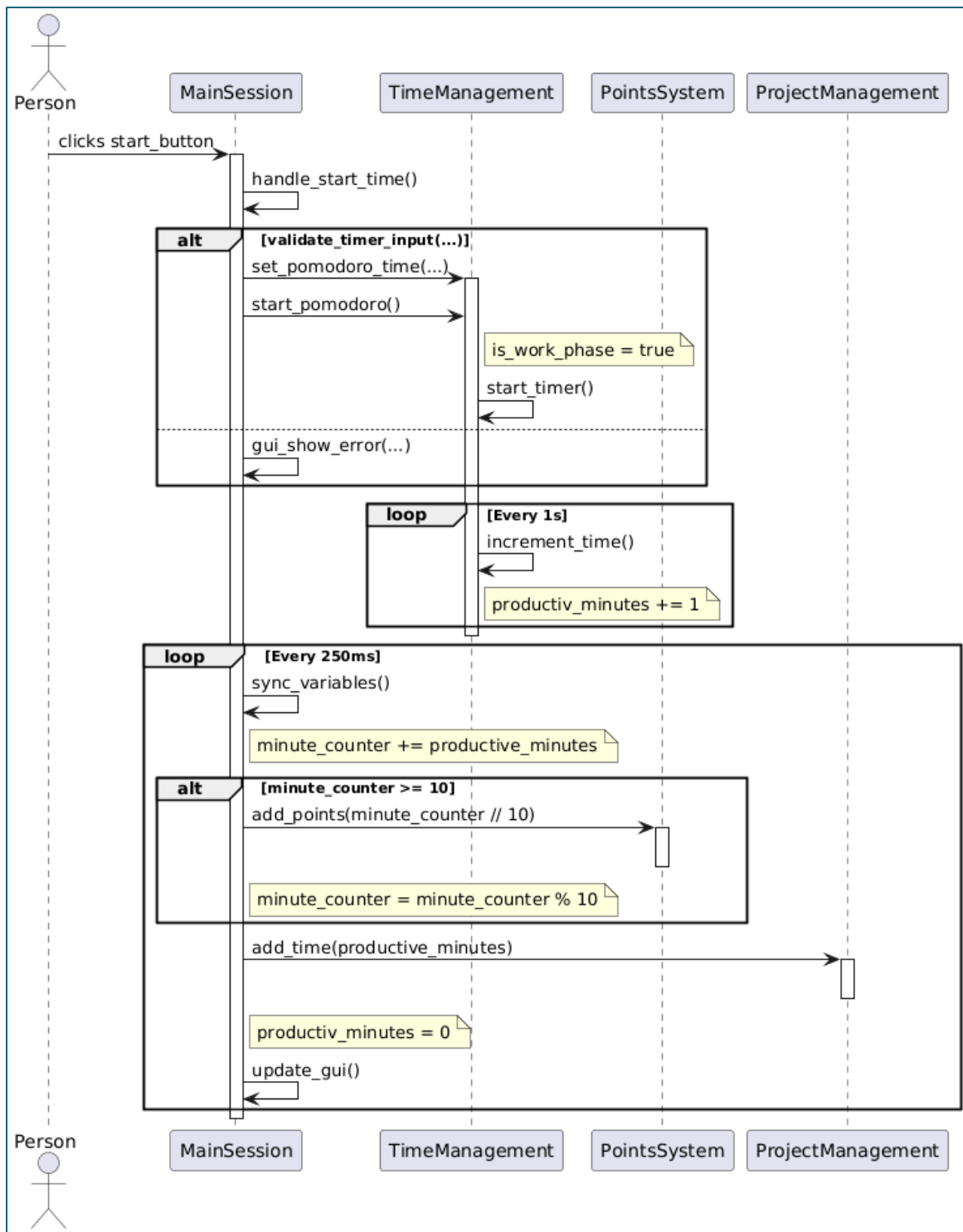


Abbildung 3 – UML-Sequenzdiagramm