Requirement Document

Alexandros Gonatas
alexthecipher@gmail.com

Alexander Spetz
spetzal@student.chalmers.se

Haoning Ma
mahaoning21@outlook.com

Xingrong Zong
xingrong@student.chalmers.se

September 2025

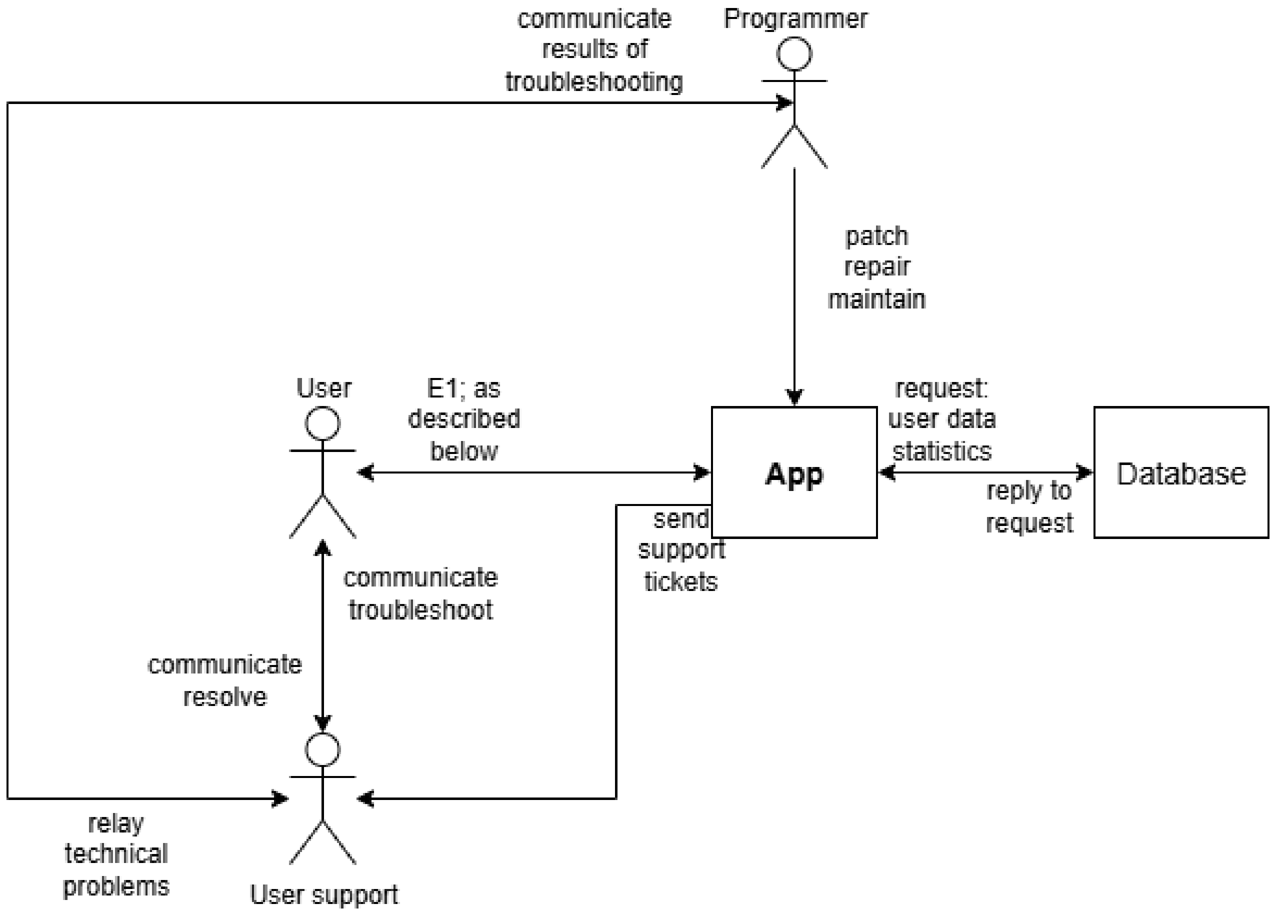
Contents

1 High-level Description

1.1 Goal and scope

The goal of the application is to help users overcome procrastination and improve productivity by turning task management into an engaging game. The user should log their daily tasks in the application, and upon completion, they are granted rewards.

The rewards can be spent to customize and take care of a virtual pet - the virtual pet should be encouraging the user to complete their tasks by reminding them of their tasks and congratulating them by giving words of encouragement, or performing dances in the background. The virtual pet should be a blank canvas that the user can customize however they want, by giving them their own, unique name, and choosing their characteristics. The app should have a store which should include items and skins for the virtual pet, as well as furniture and trinkets to customize their environment.



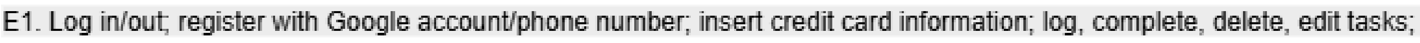

E1. Log in/out; register with Google account/phone number; insert credit card information; log, complete, delete, edit tasks;

Figure 1: Context Diagram

The context diagram (Figure 1) describes the relation of our system to other systems and agents who interact with it.

The **application** requests user data and statistics from the central database, an external system which contains this information. The **database** must reply appropriately to the system's request within a short time frame.

The **user** must register on the application with their e-mail, Google account or phone number. They can also log in/out. They can log tasks and perform these actions on them: create, edit, delete, view, complete. Once a task is completed, the user must receive rewards. They can interact with the virtual pet by customizing them, petting them; the virtual pet interacts with the user by offering them words of encouragement. Lastly, the user can spend real currency in the in-app store.

The application must ask the user for permissions: notifications, access to their calendar. The application must provide a contact center option, in case of errors or questions, which will generate a ticket for customer support agents.

The **customer support agents** must receive the tickets generated by the contact center. They must communicate with the users to ask for further information concerning their request and to update them about their ticket's status. Also, they can communicate with the programmers of the application in case of a technical error they are not able to resolve.

The **programmers** must maintain and repair the application. They can communicate with the customer support agents in case of a technical error and update them on the status of their query.

1.2 Business case and stakeholder map

1.2.1 Business Case

Perspective	Illustration	Example
Problem / Issue	Students lack self-discipline and motivation.	Many students set study goals but end up procrastinating; current apps feel "cold" as they only show numbers and checklists, leading to low engagement.
Value	Opportunities are created by this project including bridging productivity with entertainment.	Provides a **fun, AI-powered, gamified study companion** that motivates students with encouragement, achievements, and rewards for completing tasks.
Investment Gain	Financial returns and brand recognition from offering a differentiated product in a crowded market.	Early-stage monetization via in-app ads and premium subscriptions; potentially collaborate with other applications and make a sustainable profit chain.
Risks and Restraints	Possible and potential challenges and solutions.	Risk: strong competition (e.g., Forest, Habitica); Solution: differentiate with **AI personalization + emotional support**. Risk: data privacy; Solution: GDPR-compliant policies.

Nowadays, people use digital tools for all sorts of tasks: documents, diagrams, software, websites, etc. However, they often get distracted by either browsing or by the overwhelming, constant stream of information when trying to deal with a task. This leads to progress delay that often causes anxiety and procrastination for individuals, and lowers the efficiency of work for organizations.

In order to elicit real-world results and support our hypothesis, we conducted a questionnaire which was completed by 78 people. 43% of the participants answered both that they are very or extremely distracted by their phone daily and that they (very) often postpone important tasks. The correlation between phone usage and procrastination is undoubtedly high, as only 26% of the participants are not distracted by their phone and not procrastinating important tasks.

There are many productivity apps already, but in reality, people still get frustrated when they are not completing tasks in time. Is it because those apps are not helpful? A simple answer is no, since each individual reacts and progresses differently, and apps are designed to serve diverse users.

According to our questionnaire, most people postpone executing their tasks because of lack of motivation, distractions and overwhelming workload, as **Figure 2** suggests:

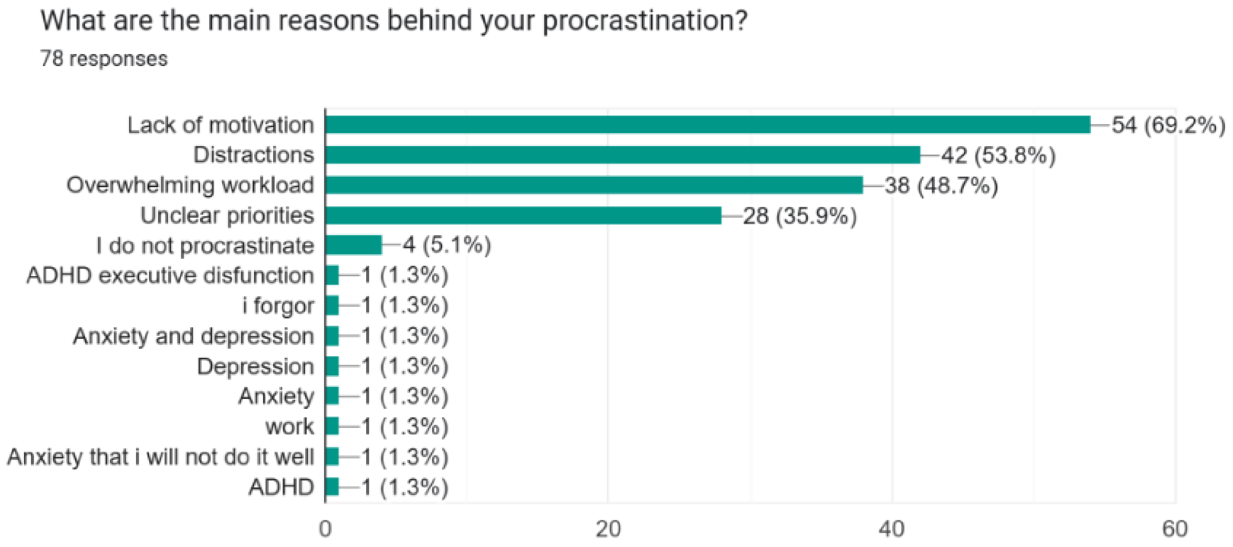


Figure 2: *Chart showing the responses to the question "What are the main reasons behind your procrastination?"*

Furthermore, only 42% of participants who have used productivity apps are still using them, which suggests a mid-low rate in the success of these apps. When asked why they are not using the apps anymore, most participants answered that they got bored/forgot about the app, or that they got frustrated from the microtransactions and paywalled content these apps had. Additionally, some participants mentioned that losing daily streaks were causing them to feel anxious or disappointed.

These metrics are suggesting that productivity apps are not useful. Yet, when the participants were asked whether they would download an app that is gamifying task scheduling and completion, 72% of participants answered positively. This statistic is indicating that people are *positive* about downloading an app to help them organize their daily life and entertain them, as long as the app remains engaging, smooth, easy to navigate and not overwhelming with monetary transactions and premium content.

With current technology trends, it will also be innovative to collaborate with Artificial Intelligence (AI) to, for example, collect user data to provide a customized, productive plan.

The expected benefits include:

- For users – improved concentration, higher task completion rate, reduced anxiety, and a fun, motivating way to cultivate long-term discipline.
- For stakeholders – stronger customer engagement, potential for workplace productivity programs, and positive impact on mental well-being.

From a business perspective, the project creates new revenue streams through premium subscriptions, in-app advertising, and B2B licensing to educational institutes or companies. It also enhances brand value by positioning us as an innovator in the digital productivity and wellness market.

Risks include intense competition, user-privacy concerns, and challenges in long-term engagement. These will be mitigated through clear market differentiation (AI-personalization + gamification), robust data-protection policies compliant with GDPR, and continuous user-experience testing to maintain high retention.

1.2.2 Stakeholders

Name	**Relationship**	**Representative**	**Sentiment/ Power**
Chalmers University of Technology	Potential adopter for B2B licensing	Department contact	Neutral / Medium
Development Team	Builddesign and maintain the system	Group members	Positive / High
Game licensing company	Market and sell the game, invest money in the game	Employee in charge of	Medium / High
People with anxiety	Users	Person with anxiety	Positive / High

Table 1: Stakeholder Map, representing potential stakeholders

Stakeholder 1 Chalmers university of technology: The university could be potential investors in the app because the development team is made up of students and the university would be interested in seeing the students succeed. They could also be interested in the app as it could boost their students productivity.
Stakeholder 2 Development team: This is the team of developers that has created and maintains the app.
Stakeholder 3 Game licensing company: A game licensing company would invest in the game which would allow the development team to develop and maintain it. The company would also make it easier of market and eventually sell the game.
Stakeholder 4 People with anxiety: According to our questionnaire, 75% of participants who reported being highly anxious in their daily lives, would use a gamified task management app.

1.3 Core functionality

UML diagram (note: has to focus on the actions performed by the APP, not the actors):

1.3.1 Actors

- User - the main actor, interacts with all core functions.
- Notification Service - secondary actor, delivers reminders/alerts.
- Payment Service (Optional) - supports in-app purchases. (Card, Swish, Klarna)

1.3.2 Use Cases

- Manage To-Dos
- Receive Reminders
- Earn Rewards
- View Progress Statistics

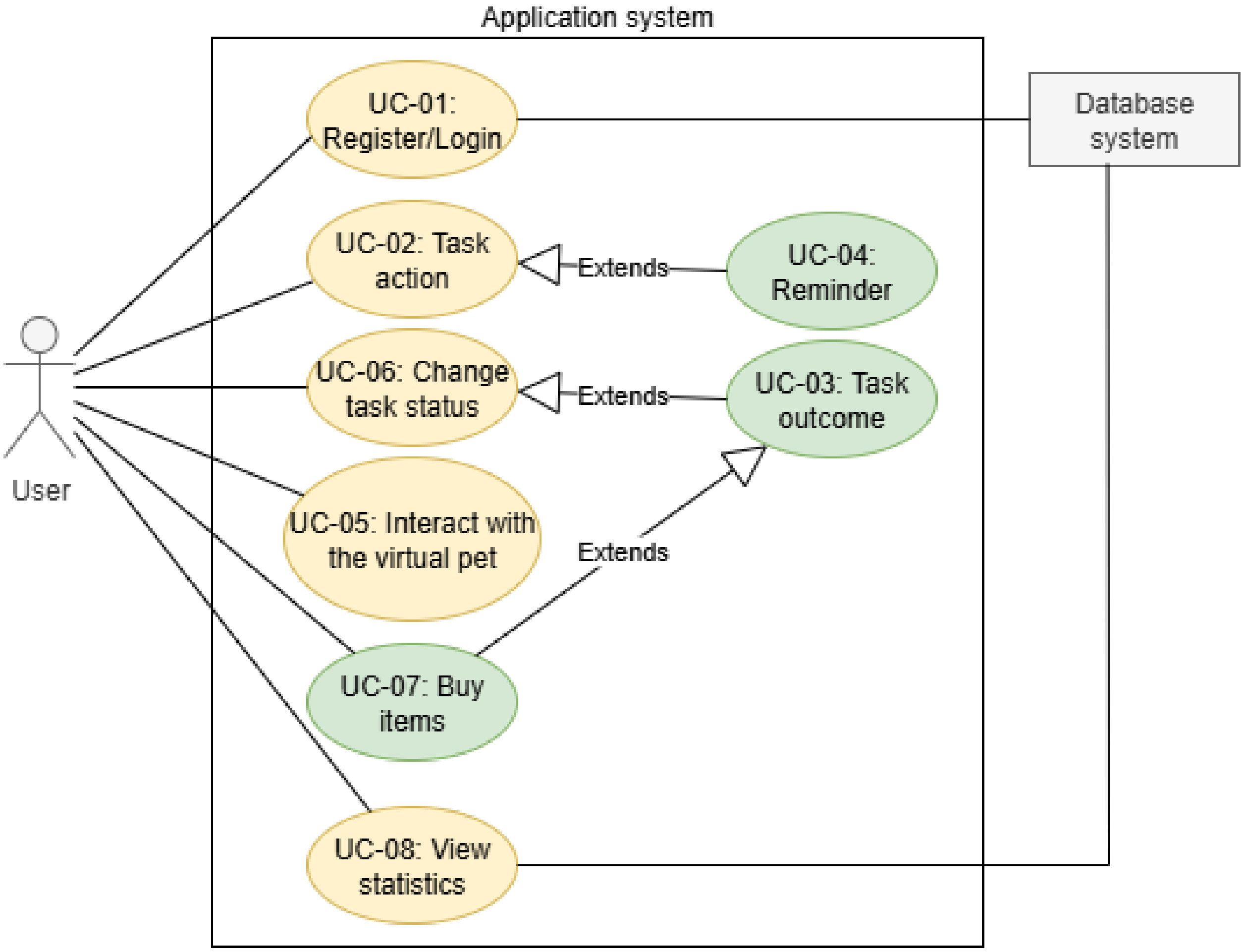


Figure 3: Use Case Diagram

UC-01: Register/Login. This initial action requires the user to either create an account or use their account to login the application. It is required as their data is unique and this is the optimal way to store them. The user's data is stored in the database, which means that the database system has to provide and verify the user's identity.

UC-02: Task action. The user can create, edit, delete, view or change the status of tasks.

UC-03: Task outcome. Each task has a reward attributed to it. If a user has missed the deadline on one of their tasks or completes one of their tasks, they will be met with missed or gained rewards.

UC-04: Reminder. The user receives a reminder for a task in the form of a notification.

UC-05: Interact with the virtual pet. The user can change the clothes and accessories of the pet, change their name and characteristics and tap on them.

UC-06: Change task status. The status of the task has changed. This triggers UC3.

UC-07: Buy items. The user can navigate to the in-app store and purchase items with their rewards from UC-03, or real currency. The items can be beneficial for the virtual pet, to buy back a lost daily streak or other.

UC-08: View statistics. The user can view their weekly/monthly statistics on task completion. The statistics are derived from the data provided by the database system.

ID	Use Case	Description
UC-01	Register/login	Adds user or logs in user
UC-02	Task action	User creates, deletes, edits or views a task
UC-03	Task outcome	A tasks either fails or succeeds
UC-04	Set Reminder	User sets a due date and notification
UC-05	Interact with game	User interacts with the game aspect somehow
UC-06	Change task status	User changes the status of task

1.4 Performance Requirements, Specific Quality Requirements, Constraints

This section will explain non-functional requirements that exist for the application. These will include security, quality and for which platforms the application will be accessible. The quality grid below is meant to explain the category some of the non-functional requirements belong to as well as their importance.

	Critical	**Important**	**Normal**	**Unimportant**	**Ignore**
Operation					
Integrity / Security			QR-06/07		
Correctness			X		
Reliability / Availability		QR-01/03			
Usability			QR-05		
Efficiency		QR-02/04			
Revision					
Maintainability			X		
Flexibility			X		
Testability			X		
Transition					
Portability			X		
Interoperability			X		
Installability			X		
Usability			X		

Table 2: Quality grid concerning quality factors

Quality targets (high level) (Estimated numbers, any feedback appreciated)

- QR-01 Task completion speed: p95 from tapping “Complete” to reward shown $\leq$ 500 ms online; $\leq$ 2s offline cache. Priority P0.
- QR-02 App start time: launch to interactive “Today” median 1.5 s, p95 2.5 s (worst 2.5 / 4). Priority P0.
- QR-03 Availability: monthly uptime for create/open/complete $\geq$ 99.5% (worst 99.0 %). Priority P0.
- QR-04 Notification on time: reminders delivered within 60 s $\geq$ 95 %. Priority P1.
- QR-05 Offline use: add/complete offline; sync after reconnect $\leq$ 10 s. Priority P1.
- QR-06 Privacy and GDPR: data export $\leq$ 48 h; account deletion $\leq$ 7 d. Priority P0.
- QR-07 Security baseline: TLS $\geq$ 1.2, encryption at rest, no secrets in client, pass baseline checks. Priority P0.

Constraints (high level):

- Platforms: iOS 15+, Android 10+, Web (latest 2 major browsers).
- Data: only what we need for core loop and subscription; analytics opt-in; no sensitive categories.
- Policies and payments: follow App Store/Play and GDPR; only subscription and cosmetic items (no cash out).

Prioritization:

- P0: QR-01, QR-02, QR-03, QR-06, QR-07.
- P1: QR-04, QR-05.

2 User Requirement Specification

2.1 Data requirements

In order to describe the data requirements of the application, an ER diagram is used to illustrate the relationships between entities that exist within the application. For this revision, the data requirements are kept at a high level, thus the diagram appears minimal. For that reason, a data dictionary was used to describe in detail the attributes of these entities.

2.1.1 ER-Diagram

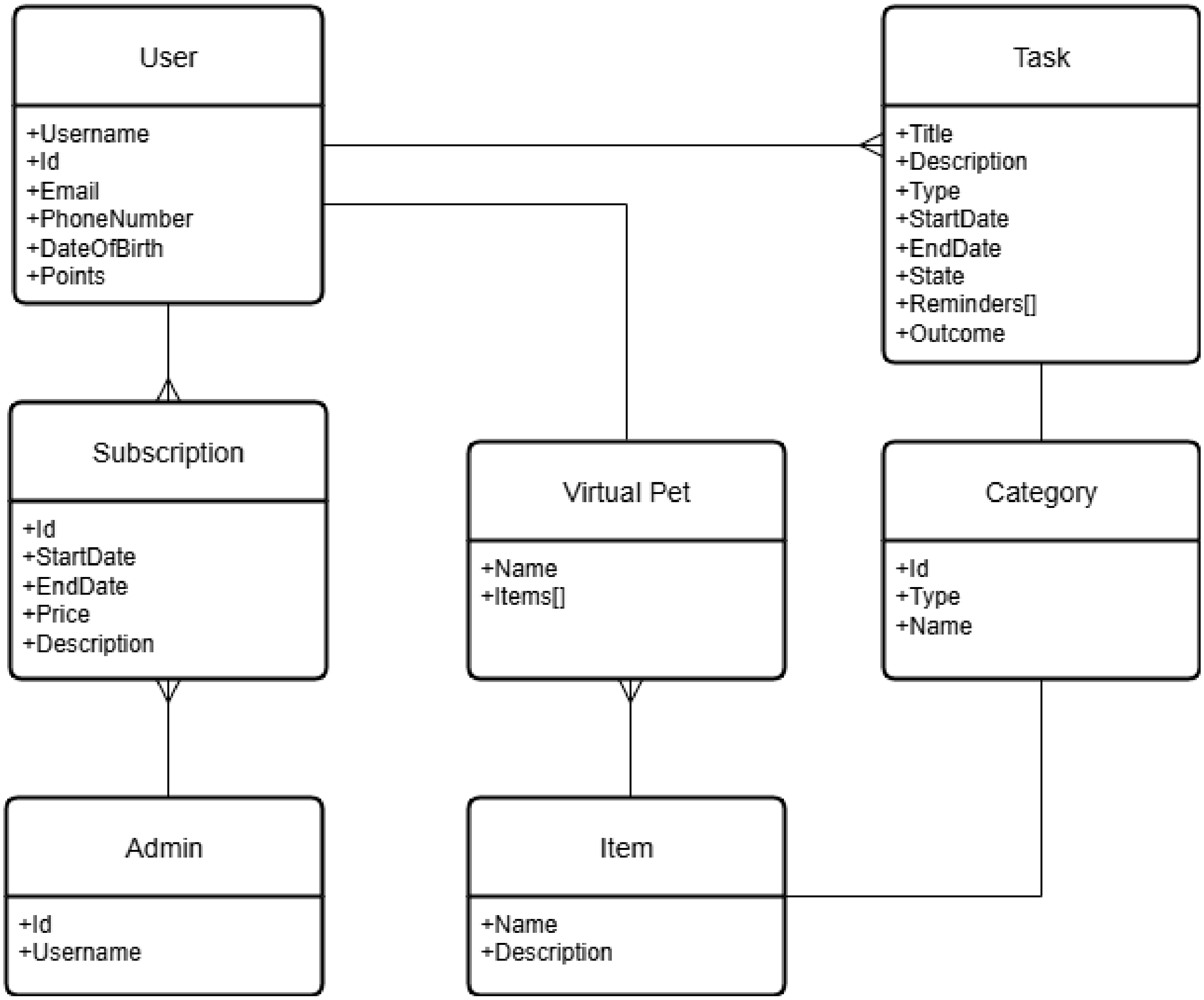


Figure 4: Entity Relationship Diagram

2.1.2 Data dictionary

Class: **User**
The ”User” class represents each user of the application. It holds the necessary logic regarding a user such as the username. The user class can also have a subscription (1:1), a virtual pet (1:1) and tasks (1:N).

Attribute	Type	Description
Id	Int	A unique id that represents each user
Username	String	A username for the user
Phone number	Tel	A provided phone number by the user
E-mail	Email	A provided e-mail by the user
Date of birth	Date	The user's birth date
Points	Int	The points the user can use to buy items for the Virtual Pet

Class: **Task**

The ”Task” class represents a task in the application. The type of a task can be attributed to a category (1:1); categories can characterize a task as a study task or a sports task, for example.

Attribute	Type	Description
Title	String	The title of the task
Description	String	A description of the task
State	Category	Contains the status of the task
Start date	DateTime	The start date and time
End date	DateTime	The end date and time
Reminder	Time Array	A user can set one or many reminders for a task
Type	Category	The type of the task
Outcome	Category	The outcome of the task

Class: **Subscription**

The ”Subscription” class represents the subscription plan a user can have.

Attribute	Type	Description
Id	Int	The unique id of the subscription plan
Price	Int	The cost of the subscription
Start date	DateTime	The date and time the user chose the subscription
End date	DateTime	The date and time the subscription expires for a user
Description	String	The description of the subscription plan

Class: **Administrator**

The ”Administrator” class represents the highest-level account responsible for the app. The Administrator provides the subscription plans (1:N).

Attribute	Type	Description
Id	Int	The id of the account
Username	String	The username of the account

Class: **Category**

The ”Category” class represents a general entity that is used to describe attributes, like the 'Outcome' of a 'Task'. The 'Type' attribute is an enum that describes the different categories; for example, it separates 'Outcome' categories from 'State' categories. This attribute will be useful for showing the correct categories in their corresponding fields in the application.

Attribute	Type	Description
Id	Int	The unique id of the category
Type	ENUM	The type of the category
Name	String	The name of the category

Class: **Virtual Pet**

The ”Virtual Pet” class is an entity representing the virtual pet that is the main feature of the game design of the app. They can be customized using items purchased with points earned by completing tasks.

Attribute	Type	Description
Name	String	The name of the pet, given by the user
Items	Item Array	Custom items that the pet can wear or hold, bought by points

Class: **Item**

The ”Item” class represents any items that can be held by, or used to customize the virtual pet. They can be purchased with the points earned by completing tasks. The category of each item refers to a general item category, such as clothing or textile.

Attribute	Type	Description
Name	String	The name of the item
Description	String	A description of what the item does
Category	Category	The category of the item

2.2 Functional requirements

This section will provide a list of tables containing user stories and the stakeholders related to them. Each table will also contains a description of the user story and a measure of how we consider a user story completed.

2.2.1 Task Creation and Management

User Story	As a user, I want to create, edit, and delete so that I can manage my workload effectively.
Description	Users can add unlimited tasks, edit their details (title, description, category, deadline), and delete them when no longer needed.
Stakeholders and Business Value	**Users:** Essential for organizing personal work; reduced friction in task entry. **Investors/Owners:** A basic requirement for user retention — without smooth task management, retention drops.
Success Measure	Users report that entering tasks feels fast and intuitive.

2.2.2 Categories and Projects

User Story	As a user, I want to group tasks into categories/projects, so that I can organize complex goals.
Description	Tasks can be tagged under categories (Work, Study, Health, etc). Filter and sort views allow quick navigation
Stakeholders and Business Value	**Users:** Helps them manage multiple areas of tasks; reduces clutter. **Investors/Owners:** Supports upselling advanced organization tools.
Success Measure	Success Measure: Users create ≥ 1 category on average in their first week.

2.2.3 Deadlines and Scheduling

User Story	As a user, I want to set deadlines and recurring tasks so that I can plan my time better.
Description	Users can assign due dates/times, create repeating tasks (daily, weekly, custom).
Stakeholders and Business Value	**Users:** Increases usefulness beyond simple lists; supports productivity habits. **Investors/Owners:** Improves user retention.
Success Measure	% of users who schedule tasks grows monthly.

2.2.4 Notifications and Reminders

User Story	As a user, I want reminders for tasks so that I don't forget what I planned.
Description	Push notifications, in-app banners for approaching deadlines or streak loss.
Stakeholders and Business Value	**Users:** Improves success rate in completing tasks. **Investors/Owners:** Increases daily active use and reduces churn.
Success Measure	Number of tasks completed on time increases after enabling reminders.

2.2.5 Reward Allocation

User Story	As a user, I want to receive points for completing tasks so that I stay motivated.
Description	Completing tasks grants points that work as an in-game currency.
Stakeholders and Business Value	**Users:** Keeps engagement high and gamifies productivity. **Investors/Owners:** Encourages premium upgrades to raise the daily rewards.
Success Measure	$\geq$ 30 % of active users engage with the reward system weekly.

2.2.6 Reward spending

User Story	As a user, I want to spend the points in something interesting, because I want to feel like my rewards matter and have impact.
Description	Using rewards to buy items with in-game currency that can be used to customize your Virtual pet.
Stakeholders and Business Value	**Users:** Causes excitement, sets personal goals, increases motivation for completing tasks. **Investors/Owners:** Differentiates app from existing task management systems, creates brand/name value.
Success Measure	New demands for the game per month, high-normal rate users spend their rewards.

2.2.7 Progress Tracking and Badges

User Story	As a user, I want to keep track of my previous result so that I can track my progress, and get rewarded for keeping up with my tasks.
Description	Creating a streak when completing tasks several times in a row, getting badges and rewards for completing a certain amount of tasks in a row and being able to see previous days completed or failed tasks.
Stakeholders and Business Value	**Users:** Motivation, clear picture of progress, measurable success/failure. **Investors/Owners:** Encourages user retention.
Success Measure	Positive user feedback for rewards, increased user engagement.

2.2.8 Analytics and Statistics

User Story	As a user, I want statistics about my task completion so that I can reflect on my productivity.
Description	Charts of completed vs. pending tasks, reward earnings, and time-to-completion.
Stakeholders and Business Value	**Users:** Supports self-improvement. **Investors/Owners:** Data provides insights for feature optimization.
Success Measure	Feedback shows statistics aid goal-setting.

2.3 Detailed Performance Requirements, Specific Quality Requirements, Constraints

Tag: GIST: A short description to help understanding ambition/PLAN: The level at which success can be claimed SCALE: The scale of measurement used to quantify the statement METER: The process or device used to measure using the SCALE MUST: The minimum level required to avoid failure STRETCH: The best if everything goes perfectly WISH: A desirable level of achievement RECORD: The best-known achievement PAST: Previous results that may be used for comparison TREND: A set of historical data or an extrapolation of this STAKEHOLDER: A person or organisation materially affected AUTHORITY: The person, group or level of authorization allocated DEFINED: The official definition of a term

The numerical targets are based on usability heuristics (Nielsen's 0.1–1 s rule), mobile platform guidelines (Google, Apple), and GDPR compliance rules. They serve as initial quality goals and will be adjusted based on empirical measurements in later iterations.

2.3.1 QR-01: Task completion speed

Tag	Performance.
Description	Time from tap “Complete” on a task to reward screen shown.
Scale	milliseconds (p95).
Goal	$\leq$ 500ms online.
Stretch	$\leq$ 300ms.
Fail	$\geq$ 2s.
Priority	P0.
Owner	Dev team.
Rationale	Responsiveness is crucial for habit-forming.

Table 3: Caption

2.3.2 QR-02: App Cold Start

Tag	Performance, Usability.
Description	Time from launching app to interactive “Today” screen.
Scale	Seconds (median and p95).
Meter	Mobile analytics logs.
Goal	Median $\leq$ 1.5s, p95 $\leq$ 2.5s.
Stretch	Median $\leq$ 1.0s, p95 $\leq$ 2.0s.
Fail	p95 $\geq$ 4s.
Stakeholder	New users, app store reviewers.
Rationale	Fast startup is tied to activation (finishing a task on day 1).

Table 4: Caption

2.3.3 QR-03: Availability of core flows

Tag	Reliability.
Description	System uptime for task creation, opening, completion.
Scale	Percentage uptime per month.
Meter	Monitoring logs (server health).
Goal	99.5 % monthly uptime.
Stretch	99.9 %.
Fail	$\leq$ 99.9 %.
Stakeholder	All users, especially paying subscribers.
Rationale	Ensures trust and user retention.

Table 5: Caption

2.3.4 QR-04: Notification timeliness

Tag	Correctness, Reliability.
Description	Reminders delivered within a certain time frame.
Scale	Percentage of reminders delivered within 60 s of due time.
Meter	Push service logs.
Goal	$\geq$ 95 %.
Stretch	99 %.
Fail	$\leq$ 85 %.
Stakeholder	Users relying on reminders, especially students.
Rationale	Critical for supporting productivity, helps avoid procrastination.

Table 6: Caption

2.3.5 QR-05: Offline Continuity

Tag	Robustness, Usability.
Description	Ability to add/complete tasks online and sync after reconnect.
Scale	Seconds to sync.
Meter	Client logs, QA test cases.
Goal	Sync $\leq$ 10 s after reconnect.
Stretch	$\leq$ 5s.
Fail	$\geq$ 20 s for data loss.
Stakeholder	Mobile users with poor connectivity.
Rationale	Core to user experience and inclusivity in unreliable network environments.

Table 7: Caption

2.3.6 QR-06: Privacy and GDPR Compliance

Tag	Privacy, Legal.
Description	Personal data management.
Scale	Hours/days to execute user requests.
Meter	Support system logs.
Goal	Data export $\leq$ 48h; account deletion $\leq$ 7d.
Stretch	Export $\leq$ 24h; deletion $\leq$ 3d.
Fail	$\geq$ 10d.
Stakeholder	Users, regulators, business.
Rationale	Critical for trust and compliance (business goal: trust).

Table 8: Caption

2.3.7 QR-07: Security baseline

Tag	Security.
Description	Protection of user data and app integrity.
Scale	Compliance with security checklist (TLS $\geq$ 1.2, encryption at rest, no hard-coded secrets).
Meter	Pen tests, static code analysis.
Goal	Pass all baseline checks.
Stretch	Achieve OWASP Mobile Top 10 compliance.
Fail	Failing any baseline check.
Stakeholder	Users, IT security team, business.
Rationale	Required to protect brand reputation and prevent breaches.

Table 9: Caption

Quality Requirements:

- QR-01 Task completion speed: milliseconds from tap “Complete” to reward/pet visible on client (p95). Target $\leq$ 500 ms online; $\leq$ 2 s offline cache; worst 1.5 s / 3 s.
- QR-02 App cold start: time to interactive “Today” (median/p95). Target 1.5 s / 2.5 s; worst 2.5 s / 4 s.
- QR-03 Availability of core flows: monthly uptime for create/open/complete. Target 99.5%; worst 99.0%; planned downtime announce $\geq$ 48 h before.
- QR-04 Notification timeliness: percent of due reminders delivered within 60 s of planned local time. Target $\geq$ 95%; default $\leq$ 3 per day; respect quiet hours.
- QR-05 Offline continuity: can add/complete without internet; sync on reconnect $\leq$ 10 s; no duplicate records.
- QR-06 Privacy and GDPR: data export $\leq$ 48 h; deletion $\leq$ 7 d (including backups per policy); store only needed fields (email, timezone); analytics opt-in.
- QR-07 Security baseline: TLS $\geq$ 1.2; encryption at rest; no hard-coded secrets in clients; pass basic mobile/web checks.

Quality scenarios:

- QS-1 Instant reward (QR-01): tap “Complete” and reward appears almost at once (p95 $\leq$ 500 ms).
- QS-2 Quick first use (QR-02): first open shows “Today” fast; user can add first task quickly (1.5 s / 2.5 s).
- QS-3 Offline then sync (QR-05): no internet, user completes tasks; after reconnect, sync finishes within 10 s.
- QS-4 Trust action (QR-06/QR-07): user requests account deletion; done within 7 days; no PII kept in logs.

2.4 Proposed prioritization

This section will discuss what we have prioritized and will prioritize for the different releases of this paper. It will also be used to advice what should be prioritized when making the app through 100-dollar test or some other form of prioritizing technique.

2.4.1 Second release

In the second release, we have:

- Add a virtual pet as the game part of the application.
- Changed functional requirements to take the virtual pet into account
- Add a streak system, meaning that completing all or most part of daily tasks for X days in a row, the user will be awarded extra rewards.
- Decide which extra rewards are appropriate.
- Done more elicitation by putting out a questionnaire.

2.4.2 Third release

In the third release we will:

- Do prioritization techniques like the 100 dollar test or Ranking.
- Try to do interviews to get more information on possible stakeholders and functional requirements.
- Try to find more requirements and stakeholders that are relevant to the application from a business perspective.

2.4.3 Future releases (optional)

In future releases, we will:

- Integrate a house/habitat/environment for the e-pet.
- Generate weekly reports and statistics for each user.
- Add a store where a user can spend their rewards to customize their e-pet and its habitat.
- Explore the option of integrating other apps that may be relevant to the application, such as the system calendar or the 'Fitness' app most mobile phones have.
- Explore a cross-platform option, in which a user can have all information available in more than one device or device type.
- Research how users would benefit from interacting with other users and their pets on the app.

3 System Requirement

3.1 System requirements

For this section, we will make use of the sections provided in the user stories to better organize the different system functions and to directly connect them to the user stories provided.

3.1.1 General

The system shall be available in iOS and Android software.
While the user is not logged in, when they open the app, the app shall prompt them to log in.
When an account does not exist, the app should prompt the user to create one.
When the user wants to buy content from the in-app store, the app shall check if the user has sufficient points to purchase the content.
When the user does not have sufficient points to buy content, the app shall prompt them to enter their credit card information and provide them the option to save the information for future transactions.
The app shall provide subscription plans to the user.

3.1.2 Task Creation and Management

The app shall provide the option to create and delete tasks, without any time constraints.
When a task exists, the app shall provide the option to edit it.
The app shall sort the registered tasks by deadline and importance.

3.1.3 Categories and Projects

The system shall provide default categories for organization purposes, such as 'Home tasks' or 'School tasks'.
While tasks have been created, the app shall make the tasks selectable and sortable.
The app shall have editable categories.
The app shall have a 'Projects' section/screen.

3.1.4 Deadlines and Scheduling

The task shall have reminders, a 'Do not repeat' option, a start date and time and an end date and time.
When the aforementioned parameters have been set in a task, the task shall be editable for those parameters.
The app shall provide the user the option to duplicate/copy a task.

3.1.5 Notifications and Reminders

While a task exists, when one or more reminders have been set, the system shall notify the user of the task.
While a reminder exists for a task, the system shall provide the option to silence the reminder.
While a reminder exists for a task, when the user's device has been set to 'Do not disturb', the system shall notify the user without system sounds.

3.1.6 Reward Allocation

While a task exists, when the user completes it, the app shall grant the user a reward.
The app shall have different rewards for different tasks.
The app shall compute the appropriate reward based on the repeatability of the task, how far in the deadline the task was completed, and the duration of the task.
While a task exists, when the user completes it, the virtual pet shall encourage the user.

The virtual pet shall provide a variety of forms of encouragement.
The app shall have a streak system.

3.1.7 Reward Spending

The app shall have a store.
The store shall provide items for the virtual pet and their habitat.
The store shall include buying points with actual currency.
While the user has a streak, when they miss a day and the streak breaks, the store shall include the option to buy back the user's streak.

3.1.8 Progress Tracking and Badges

While the user has set repeating tasks, when they complete them, the app shall award appropriate badges to the user.
The app shall use Artificial Intelligence (AI) in order to produce customized badges for the user.

3.1.9 Analytics and Statistics

The app shall provide statistics to the user.
While the user has been using the app for a month or more, the app shall provide them with a chart highlighting their monthly progress.
The app shall compute and draw charts based on the aforementioned metrics.

3.2 UI prototype

The UI prototype will be meticulously illustrated in the next release of the project.

3.3 Detailed data requirements

3.3.1 Data Constraints and Properties

Class	User
Id	Integer, unique, auto-increment, primary key.
Username	String, required, unique, 3-30 characters, no special symbols except underscore.
Phone number	Optional, must follow international format.
E-mail	Required, unique, must follow valid email pattern.
Date of birth	Optional, must ensure user is older than legal age requirement for app usage.
Relationships	One User → Many Task (1:N) One User → One Subscription (1:1) One User → One Vpet (1:1)

Table 10: User Class

Class	Task
Title	String, required, max 100 characters.
Description	Optional, max 500 characters.
State	ENUM (Pending, In Progress, Completed, Missed).
Start date	Required, must be earlier than End date.
End date	Optional, but if set, must be ≥ Start date.
Reminder	Optional, allows multiple reminders; must be later current date/time.
Type	ENUM (Study, Work, Exercise, Personal, Other).
Outcome	ENUM (Success, Failure, Partial, Skipped).
Relationships	Many Task → One User (N:1). Many Task → One Category (N:1).

Table 11: Task Class

Class	Subscription
Id	Integer, unique, primary key.
Price	Positive integer, must be in app's currency (e.g., EUR/USD/SEK).
Start date	Required, must be current or future date.
End date	Required, must be later than Start date.
Description	String, max 200 characters.
Relationships	One Subscription → Many User (1:N). One Subscription → One Administrator (1:1 ownership).

Table 12: Subscription Class

Class	Administrator
Id	Integer, unique, primary key.
Username	String, required, unique.
Relationships	One Administrator → Many Subscription (1:N).

Table 13: Administration Class

Class	Category
Id	Integer, unique, primary key.
Type	ENUM (TaskState, TaskType, TaskOutcome).
Name	String, required, max 50 characters.
Relationships	One Category → Many Task (1:N).

Table 14: Category Class

Class	Vpet
Name	String, required, max 30 characters.
Attributes (optional extension)	Level: Integer, default 1, max 100. Experience points (XP): Integer, increases when user completes tasks. Mood: ENUM (Happy, Neutral, Sad), influenced by user performance.
Relationships	One Vpet → One User (1:1).

Table 15: Vpet Class

3.3.2 Data Dictionary

Attribute	Description	Constraints
User.Username	Unique identifier chosen by user	Required, 3–30 chars, only letters, numbers, underscore
User.Email	User email for login and recovery	Required, valid email format, unique
Task.State	Current status of a task	ENUM: Pending, In Progress, Completed, Missed
Task.Start date	Start time of task	Must be earlier than End date
Subscription.Price	Cost of plan	Positive integer
Vpet.Level	Growth indicator of virtual pet	Integer, default = 1, max = 100

Table 16: Data Dictionary

3.4 Acceptance tests (optional)