

Design Document – Sneaky Golem

Start date	11/11/2025
End date	
Sprint 0	
Group	Sneaky Golem
Student Names	Boaz Vaneveld, Rick de Cuijper, Bart Geijtenbeek, Valentijn van Grunsven, Xander UijtdeHaag
Topic	Sustainable Schoolyard Maintenance
Doc version no.	1

Table of Contents

1. SYSTEM PROBLEM DOMAIN & CONTEXT	2
2. TARGET AUDIENCE & PERSONA	3
3. SYSTEM REQUIREMENTS	5
4. HIGH FIDELITY PROTOTYPE	7
5. USE CASE DIAGRAM	12
6. ENTITY RELATIONSHIP DIAGRAM	13
7. ARCHITECTURAL COMPONENT DIAGRAM	14

{Important Note: The 3 x Use Case Specifications, 3 x Activity Diagrams for those Use Cases, 3 x Data Flow Diagrams for those Use Cases (Context level) will be put into a 3 separate Sprint Reports during the course of the project, so are excluded from the first kick-off week.

At the end of the project, you need to insert these into 3 new Chapters (Use Case Specification, Activity Diagrams, Data Flow Diagrams) at the end of your final Design Document document before handing in.}

1. System Problem Domain & Context

{Introduce the document and describe the main context and challenges/opportunities that the system will address (high level link to main elements from the Design Thinking phase – reducing CO2 etc, target audience etc). Introduce the various sections, models used for the rest of the document, to guide the reader.

This document will describe the Sustainable Schoolyard Maintenance Organizer created by our group. The context surrounding this project is the following. This is a school project created at the request of the client IVN. We were tasked to create an ICT solution to one of the problems that might present itself during the creation or maintenance of a Sustainable Schoolyard. Using the Design Sprint format, we made the decision to focus on the maintenance aspect of the assignment. Our design is focused on solving the challenge of the organization of tasks and motivation of students to complete these tasks. We took great care to examine the various aspects of the current process the school had in place to curate the maintenance of the schoolyard, as well as how it tried to involve the students in this task, and tried to come up with an ICT solution that would improve and streamline these processes. We examined our target audience and determined who would benefit most from a potential new tool and how we could best fit it to their needs.

The problem we identified during the Design Thinking phase centers around the lack of structure and engagement in the current schoolyard maintenance process. Although schools have the intention to involve children in taking care of their outdoor spaces, this often happens in an unplanned or inefficient way. Tasks are unevenly distributed, teachers have limited time to organize activities, and students can easily lose interest when there is no clear system of responsibility or reward. This results in inconsistent maintenance, missed learning opportunities, and a low sense of ownership among students. Our system addresses this by providing a structured task planner combined with a motivational layer through gamification, allowing both teachers and students to take part in the maintenance process in a more organized, meaningful, and enjoyable way.

Now that the context surrounding our problem and solution has been clarified. We will go further into depth about various aspects of the process and environment. Section 2 describes our target audience in greater detail and shows off some personas we have created to use as a stand in for people that might be a part of that audience. Section 3 describes some of the requirements we have determined to fit our project. Section 4 shows the initial prototype we used to test some of our first ideas and concepts. Section 5 shows a Use Case Diagram, detailing who might interact with our created system and in what ways. Section 6 shows an Entity Relationship Diagram, displaying what kind of relation some of the information and actors have. Section 7 shows an Architectural Component Diagram, which shows how we currently have our system designed.

2. Target Audience & Persona

{Explain your target audience leading to persona and insert the persona}

We had decided to focus our efforts on the teachers and students, we came up with these audiences through these actions. First, we visited a school to figure out how they managed their schoolyard right now. From this we learned that it was mostly Teachers and students that did the work and sometimes a parent would volunteer. The kids could only do light work and follow the teacher's instructions. Teachers had to manage the what the kids did and do the heavier work. The school we visited had one teacher who knew what to do to maintain the schoolyard, so the work was done quite inefficiently. One teacher effectively had to oversee all the children.

During the design week we had decided to make a planning and organisation tool for teachers and students. We decided not to include the parents, because it was difficult to assign them tasks while keeping it voluntary.

Persona 1:

Name: Marjolijn Jansen

Age: 34

Occupation: Elementary School Teacher

Education: Bachelor's degree in education

Background

Sarah teaches a class of 25 students aged 8–10. Her school emphasizes hands-on learning and outdoor education. She enjoys involving her students in environmental and community projects, such as sport, gardening and nature observation. Marjolijn often manages class projects that extend beyond the classroom, coordinating with other teachers and parents.

Personality

Engaging and creative: Loves finding ways to make lessons fun and engaging.

Supportive: Encourages teamwork and responsibility among students.

Digitally inapt: is not very good at using digital applications.

Goals

Use the schoolyard as an interactive learning environment.

Integrate real-world environmental lessons into the curriculum.

Connect students with nature.

Challenges

Limited time during the school day for outdoor or non-lesson tasks.

Keeping students engaged and motivated in repetitive tasks.

Managing different age groups' abilities and attention spans.

Needing to clearly communicate between teachers.

Not very good at using new technology.

Motivations

Wants students to understand the value of teamwork and environmental care.

Believes that hands-on learning builds stronger connections to the community.

Persona 2:

Name: Tim de Vries

Functional Design Specification

Age: 9

Occupation: Student (Grade 4)

Education: Elementary School

Background

Tim is an energetic and curious 9-year-old who enjoys being outdoors more than sitting behind a desk. He loves playing soccer during break, exploring the schoolyard, and playing with classmates. While he sometimes struggles to stay focused during long lessons, he thrives when tasks are interactive and playful.

Personality

Curious: Always asks questions and enjoys discovering how things work.

Active: Prefers physical, hands-on activities over reading or writing tasks.

Sociable: Likes working in groups and helping friends.

Easily distracted: Needs engaging and visually stimulating activities to stay focused.

Goals

Have fun and play with friends.

Earn recognition or rewards (points, badges, praise from teachers).

Challenges

Gets bored with repetitive or purely instructional tasks.

Forgets instructions easily if they are not clear or visual.

Can lose motivation if feedback or rewards aren't immediate.

Has limited experience with structured digital learning/ organising tools.

Motivations

Enjoys games, challenges, and friendly competition.

Likes being praised by teachers or showing progress to parents.

Likes earning points or unlocking achievements.

3. System Requirements

{Summarise/explain and insert your detailed, prioritised requirements here or update your Excel-file and make a reference to the separate excel file. Also include the following requirements}

Project Requirements (PR)

MUST

- PR1. The project builds-upon the outcomes of the previous Design Thinking phase.
- PR2. The project is to be completed in three sprints of two weeks after the first kick-off sprint week.
- PR3. The application should give back the CO2 reduced and used the suggested route and JSON format

Non-Functional Requirements (General Platform)

MUST

- NF1. The technology will be based on Sveltekit for the frontend and NodeJS/Express for the backend
- NF2. The system shall support at least two microservices and a gateway
- NF3. The principles of Authorization and Authentication are out-of-scope of this project due to complexity

Functional System Requirements (FR)

MUST Have

- FR1. The system shall allow users to create new tasks with a title, description, date, timeframe, gold value, progress, and state.
- FR2. The system shall allow users to create sub-tasks from existing tasks
- FR3. The system shall allow users to edit existing tasks.
- FR4. The system shall allow users to delete tasks.
- FR5. The system shall be able to display tasks in a calendar view.
- FR6. The system shall show the total progress of all tasks, including a progress bar.
- FR7. The system shall be able to change the status of a task to Not Started, In Progress, or Completed.
- FR8. The system shall allow users to add children.
- FR9. The system shall allow users to assign children to tasks.
- FR10. The system shall be able to show which children are linked to each task.
- FR11. The system shall automatically award coins to assigned children when a task is completed.
- FR12. The system shall be able to display the coin balance for each child.
- FR13. The system shall assign an avatar to each child.
- FR14. The system shall allow users to change avatar settings (skin, hair, clothing, colors).
- FR15. The system shall allow users to purchase items with coins.
- FR16. The system shall store purchased items.
- FR17. The system shall display all avatar items that a child can purchase.
- FR18. The system shall show error messages when required fields are missing when creating or editing tasks or children.
- FR19. The system shall have the ability to register or log in

SHOULD Have

- FR20. The system shall be able to display tasks on the treasure hunt map.
- FR21. The system shall be able to display available avatar items.
- FR22. The system shall display an overview page with tasks, progress, and the map view.

Non-Functional System Requirements (NFR)

Performance (MUST)

- NFR1. The system must be able to load a task list within two seconds.
- NFR2. Changes to tasks must be visible to the user immediately.

Usability (MUST)

- NFR3. The interface must be suitable for use by children.
- NFR4. The application must be fully usable on phones and tablets.
- NFR5. Visual elements must be clear and easily recognizable.

Security (MUST)

- NFR6. Only authorized users may modify tasks and child data.

Reliability (MUST)

NFR7. The system must be at least 99 percent available during school hours.

NFR8. The system may not lose data in case of incorrect actions or technical problems.

Maintainability (SHOULD)

NFR9. The code must be modular so that components such as avatar items can easily be expanded later.

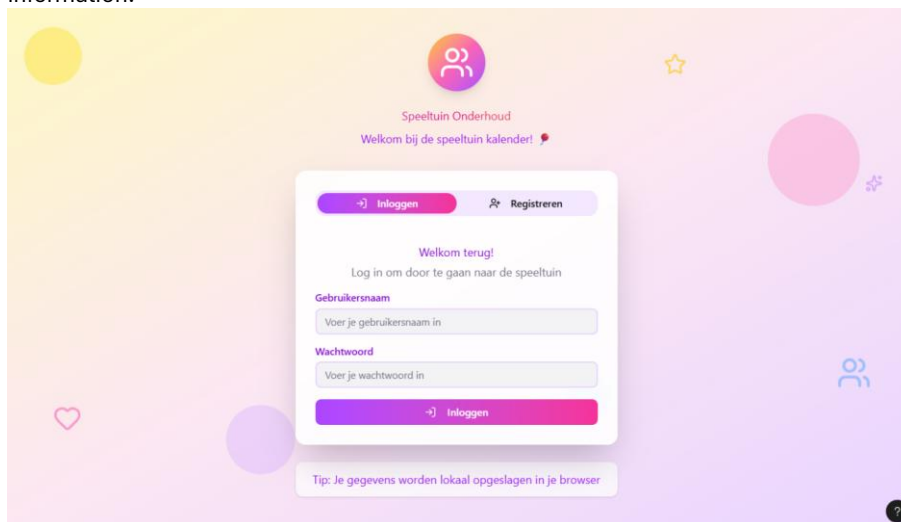
NFR10. The database must be extendable without major changes to the existing current database.

4. High Fidelity Prototype

{During the design sprint you created the first version of your prototype; In sprint 0 you created the final version of your design. Include the most important pages here. If it is a really large design; include a pdf.}

When a teacher or student opens the website, they are greeted by a simple login screen. After entering their credentials, they are taken to the main dashboard.

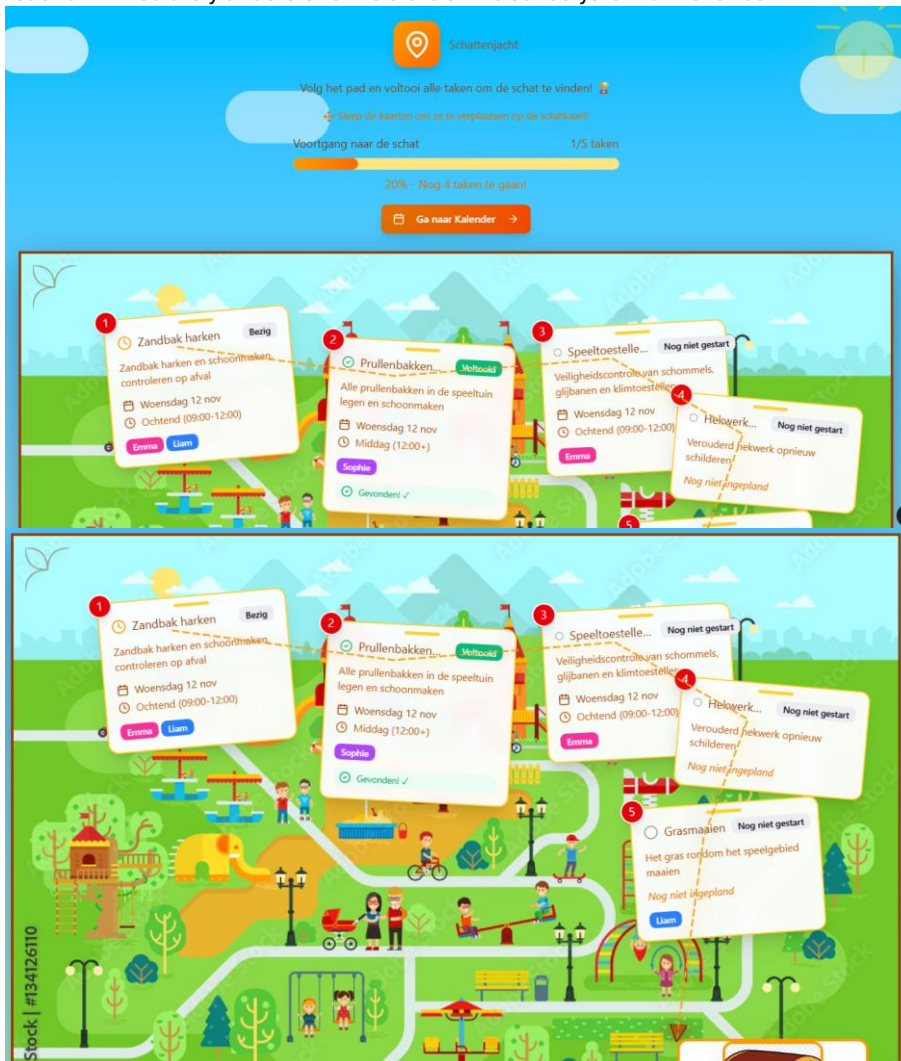
This ensures that only authorized staff members can access student and task information.



The dashboard provides a central starting point.
From here, the teacher can:

- View all upcoming tasks
- Check progress on active tasks
- Navigate to the map and avatar store
- Access children management

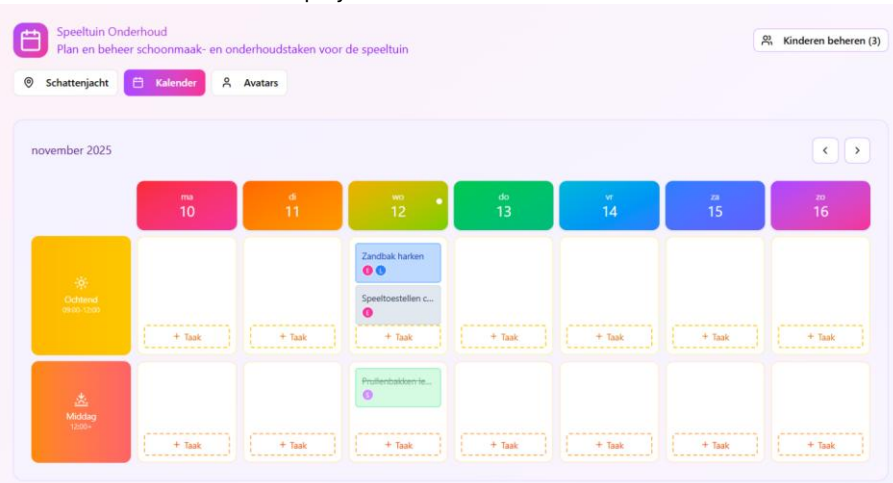
A progress bar shows how much work has been completed across all tasks, helping the teacher immediately understand the state of the schoolyard maintenance.



Teachers and students can see where work is happening:

- The **Calendar View** shows all tasks by day
- The **Treasure Hunt Map View** displays tasks visually in the schoolyard

This combines structure with play — students can “hunt” for tasks like treasure.

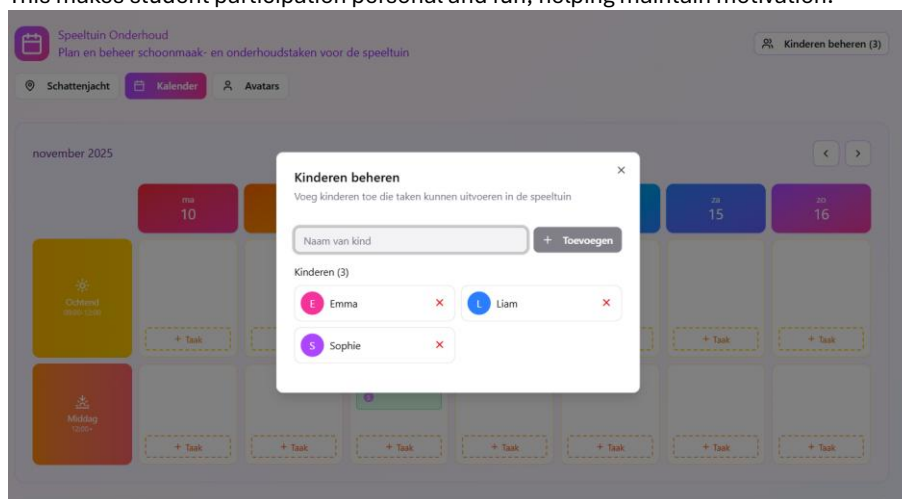


Before tasks can be planned, students must be added.

In the **Children** section the teacher can:

- Add individual children
- View existing students
- Assign each child an avatar automatically

This makes student participation personal and fun, helping maintain motivation.



From the **Task Creation Screen**, the teacher can fill in:

- Task title and description
- Date and time window
- Gold reward
- Optional subtasks
- Status (Not Started, In Progress, Completed)

Tasks can also be edited or deleted later, allowing plans to change as needed.

The screenshot displays the 'Task Creation Screen' of a task management application. At the top, there's a section titled 'Nieuwe Taken' (New Tasks) with a '+ Nieuwe taak' button. Below this, two task cards are visible: 'Hekwerk schilderen' (Painting the fence) and 'Grasmaaien' (Mowing the lawn). A tip at the bottom suggests using the calendar to plan tasks. The main part of the screen shows a calendar for November 2025. A modal window is open for creating a new task, with fields for 'Titel' (Title), 'Beschrijving' (Description), 'Subtaken' (Subtasks), 'Tijdslot' (Time slot), 'Status' (Status), and 'Toegewezen aan' (Assigned to). The 'Toegewezen aan' field shows three users: Emma, Liam, and Sophie. The modal has 'Annuleren' (Cancel) and 'Toevoegen' (Add) buttons.

Nieuwe Taken

+ Nieuwe taak

Hekwerk schilderen
Verouderd hekwerk opnieuw schilderen
Nog niet gestart

Grasmaaien
Het gras rondom het speelgebied maaien
Nog niet gestart

Tip: Sleep taken naar de kalender om ze in te plannen op een specifieke dag en tijd

november 2025

Maak een nieuwe taak aan voor de geselecteerde datum.

Titel
Voer taaknaam in

Beschrijving
Voer taakbeschrijving in (optioneel)

Subtaken
Voeg subtaak toe...

Tijdslot
Ochtend (09:00 - 12:00)

Status
Nog niet gestart

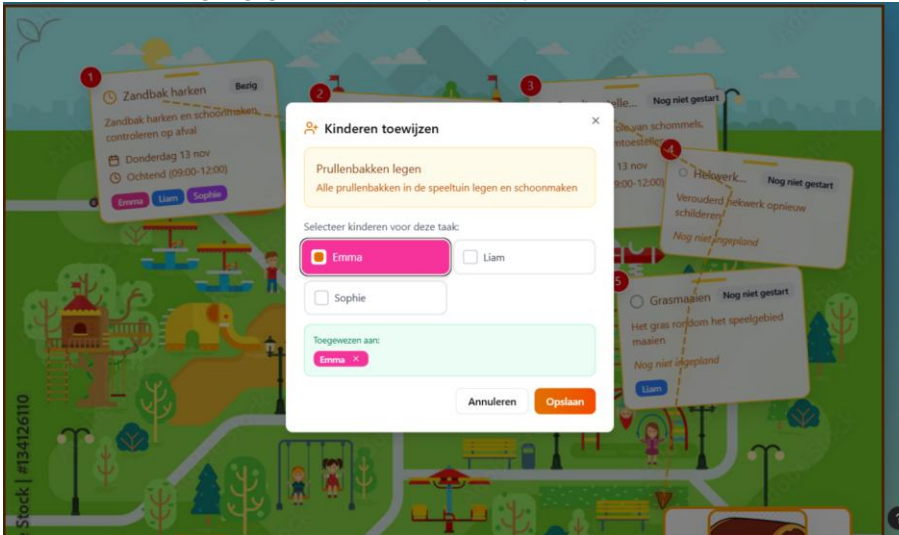
Toegewezen aan
Emma Liam Sophie

Annuleren Toevoegen

Once tasks exist, the teacher can assign students to them.

The prototype allows users to select one or more children for a task, showing clearly who is responsible.

When a task is marked **Completed**, coins are automatically awarded to the assigned children, reinforcing engagement and responsibility.

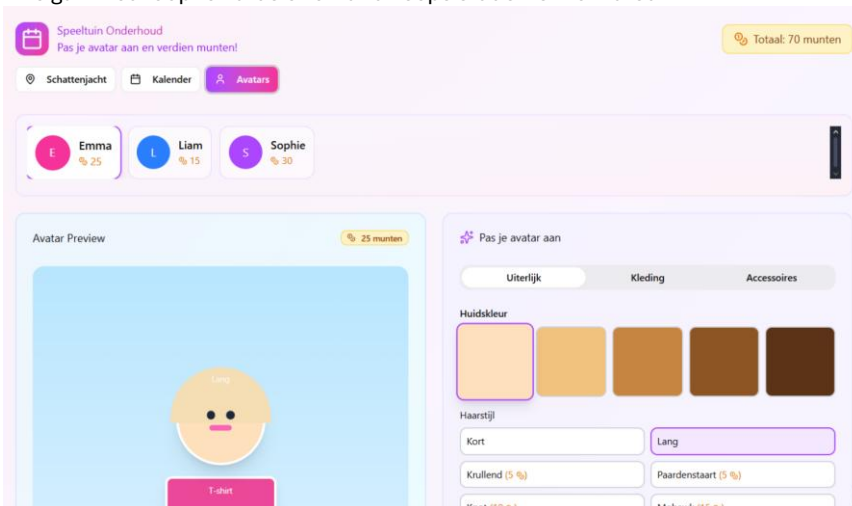


After earning coins, children can spend them.

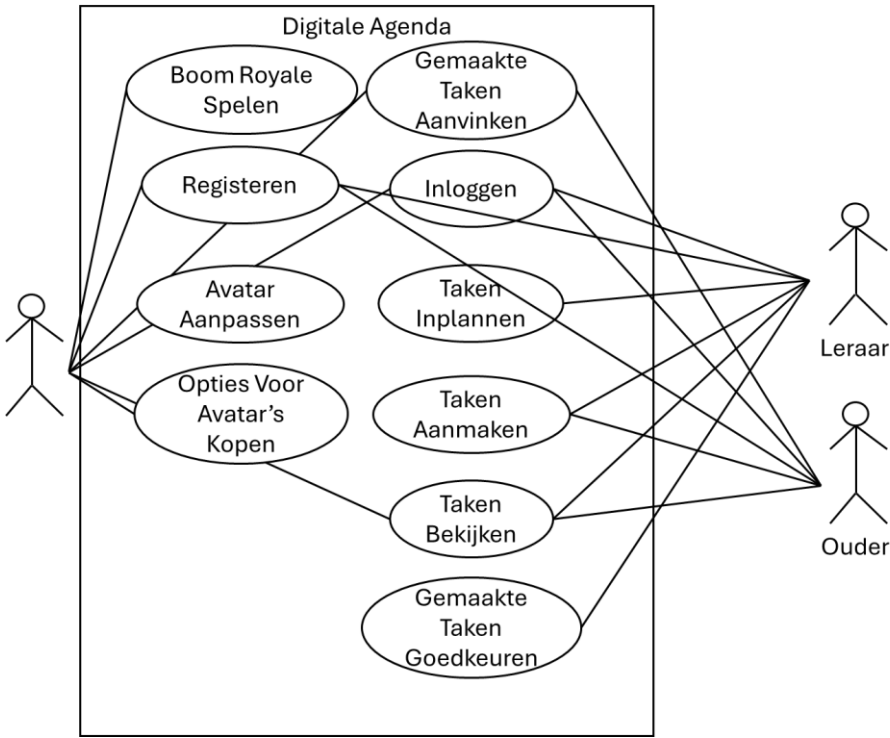
In the store, students can:

- View available avatar items
- Purchase clothing, hair, or accessories
- Equip items they own
- See their coin balance update

This gamified loop rewards effort and keeps students motivated.

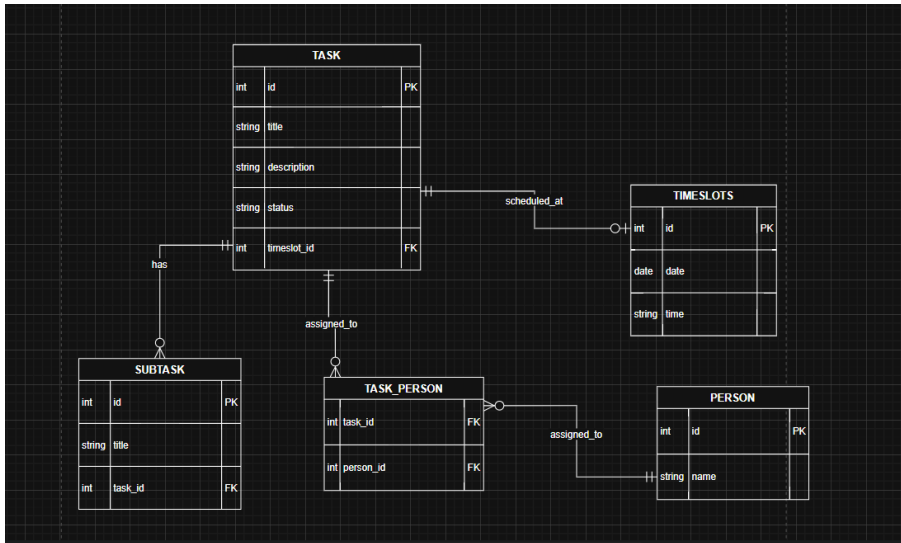


5. Use Case Diagram



6. Entity Relationship Diagram

{insert your diagram here, or link to a separate .pdf file, and explain it; visual paradigm is a great tool to create this diagram}



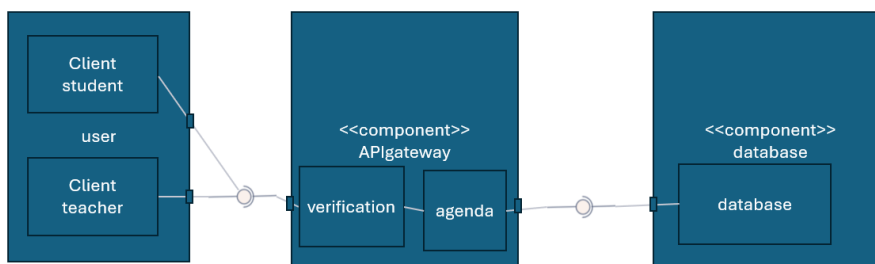
Met opmerkingen [ME1]: Zero explanation yet there is a huge jump between a UCD and this. Also consider a Logical Data Dictionary to define entities and attributes better. Why is there an AvatarID in Avatar if there is a 1:1 relationship with Child? Why is there no link between Avatar and AvatarItem – maybe okay but looks weird. Looks missing some things – such as a list of Items for a specific Avatar or are they all generic for all Avatars? This confusion is why explanations are needed.

Should tasks have categories? Are these in-built tasks (sort of 'standard tasks') or are they all 'custom' tasks? If the latter isn't that not so efficient from a UX perspective. If the first then why isn't there a task type or category? Timeframe – what is that? Isn't it sort of covered by DueDate? The link between GoldValue and GoldCoins etc. not clear what they are / it is so whether it is in the right place etc. All looks a little too simplistic but may not be a problem....!?

7. Architectural Component Diagram

{insert your diagram here and explain it; visual paradigm is a great tool to create this diagram}

Any user (student or teacher) connects to the API gateway, the gateway connects to the database. Everything goes through verification which checks who is who. The agenda processes the information and either stores it in the database or sends info back to the user.



Met opmerkingen [ME2]: This is looking okay but is incomplete – where is the API Gateway and where is the Front-End Device(s) accessing the APIs? I see 3 clients on the side but they should actually be in component form too What is the account End point connecting too. Again zero explanation so lots of questions....