SENG 383
FINAL PROJECT
REPORT

Prepared By:

Zeynep Nur KARABAY 202127011

Mert BURSALIOĞLU 202227004

December 2025

CONTENTS

1- Project Metadata & Access (Repository Access)

Project A: BeePlan (Course Schedule Generator)

- **Developer:** Zeynep Nur (Student A - Gemini)
- **Role:** Backend Algorithm & Web Integration (Python/Flask)
- **GitHub Repository Link: https://github.com/ZNurK/Seng383/tree/beeplan/BEEPLAN**
- **Repo Content Check:**
    - /src: app.py, scheduler.py, models.py
    - /docs: Diagrams & Reports
    - /video: Final Presentation Video
    - README.md: Installation instructions (Python 3.7+, Flask)

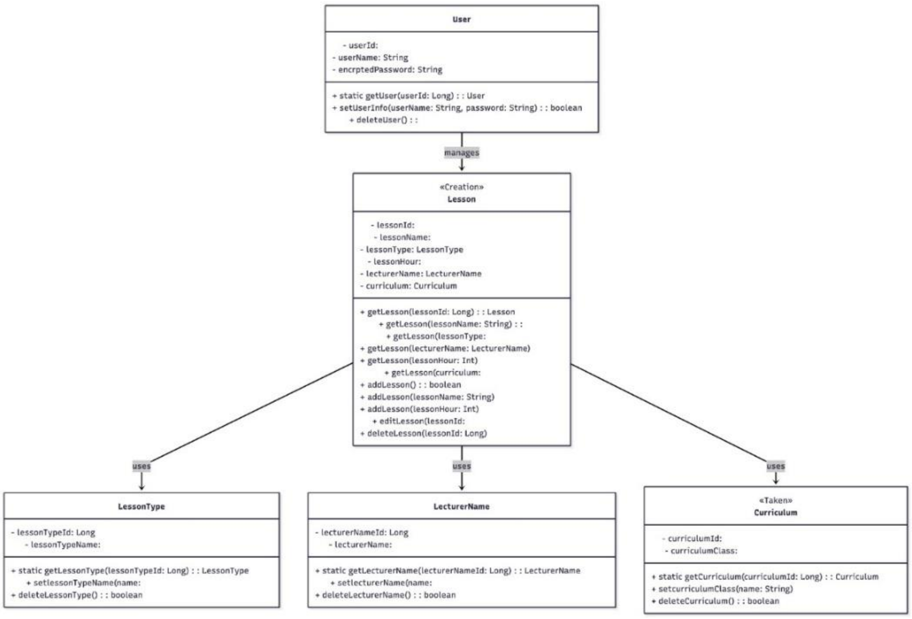
Project B: KidTask (Task & Wish Management System)

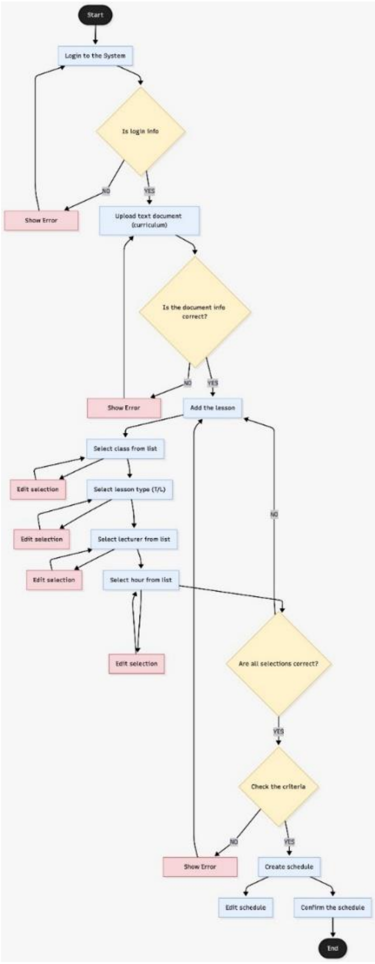
- **Developer:** Mert Bursalıoğlu (Student B - Claude)
- **Role:** Java Spring Backend & Architecture
- **GitHub Repository Link:** https://github.com/ZNurK/Seng383/tree/main
- **Repo Content Check:**
    - /src: Java Spring Boot Source Codes (com.taskmanager.*)
    - /docs: Architecture Diagrams
    - /video: Demo Video
    - README.md: Maven build instructions (mvn spring-boot:run)

2. Design Artifacts: Final Versions

This section details the final architectural designs for both systems.

2.1. BeePlan (Python/Flask) Design

- **Class Diagram:**
    - **Classes:** Instructor (availability), Course (theory/lab linkage), Classroom (capacity), Scheduler (backtracking logic).
    - 
User
- userId:
- userName: String
- encrptedPassword: String
+ static getUser(userId: Long) : : User
+ setUserInfo(userName: String, password: String) : : boolean
+ deleteUser() : :
manages
«Creation»
Lesson
- lessonId:
- lessonName:
- lessonType: LessonType
- lessonHour:
- lecturerName: LecturerName
- curriculum: Curriculum
+ getLesson(lessonId: Long) : : Lesson
+ getLesson(lessonName: String) : :
+ getLesson(lessonType:
+ getLesson(lecturerName: LecturerName)
+ getLesson(lessonHour: Int)
+ getLesson(curriculum:
+ addLesson() : : boolean
+ addLesson(lessonName: String)
+ addLesson(lessonHour: Int)
+ editLesson(lessonId:
+ deleteLesson(lessonId: Long)
uses
uses
uses
LessonType
- lessonTypeId: Long
- lessonTypeName:
+ static getLessonType(lessonTypeId: Long) : : LessonType
+ setlessonTypeName(name:
+ deleteLessonType() : : boolean
LecturerName
- lecturerNameId: Long
- lecturerName:
+ static getLecturerName(lecturerNameId: Long) : : LecturerName
+ setlecturerName(name:
+ deleteLecturerName() : : boolean
«Taken»
Curriculum
- curriculumId:
- curriculumClass:
+ static getCurriculum(curriculumId: Long) : : Curriculum
+ setcurriculumClass(name: String)
+ deleteCurriculum() : : boolean
- **Activity Diagram:**
    - **Flow:** Admin Login -> Upload JSON Data -> **AI Scheduling Algorithm** -> Conflict Check -> View Timetable -> Export CSV.



GUI Screenshots:

- **Left:** Initial Console Design.
- **Right:** Final Web Dashboard (Color-coded Timetable).

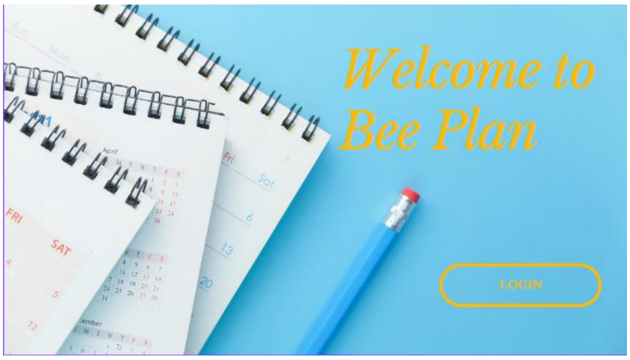


Figure 1:This screen provides a simple and clean welcoming page.

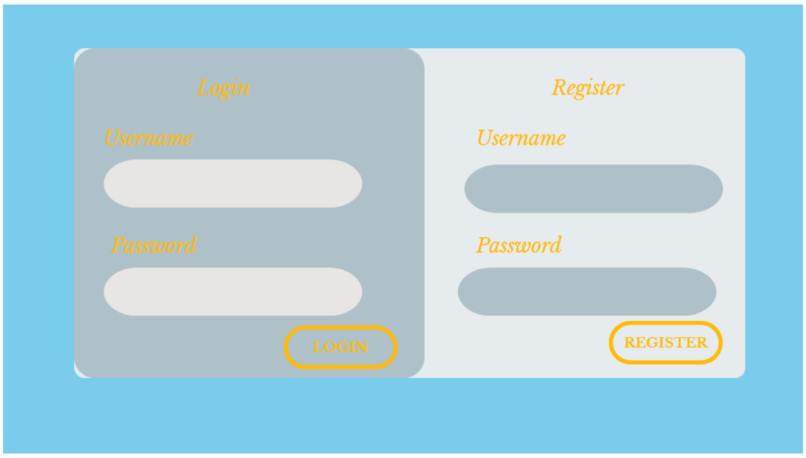


Figure 2: This screen provides a simple and clean authentication interface where users enter their username and password to access the system.

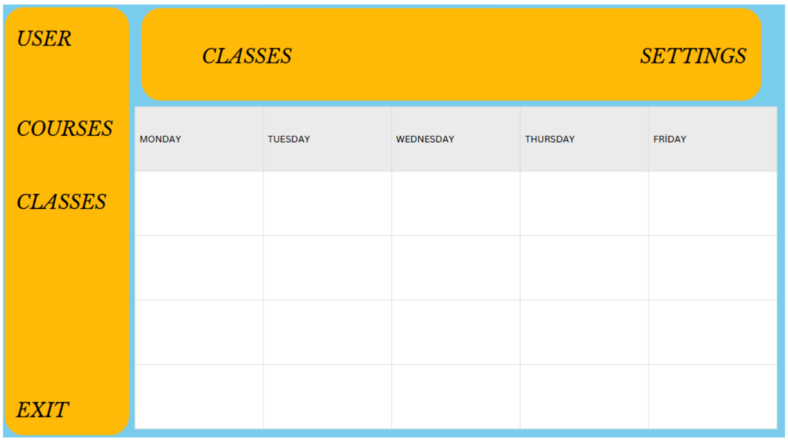


Figure 3: The dashboard provides all system functions in a card-based structure for easy navigation and visual clarity.

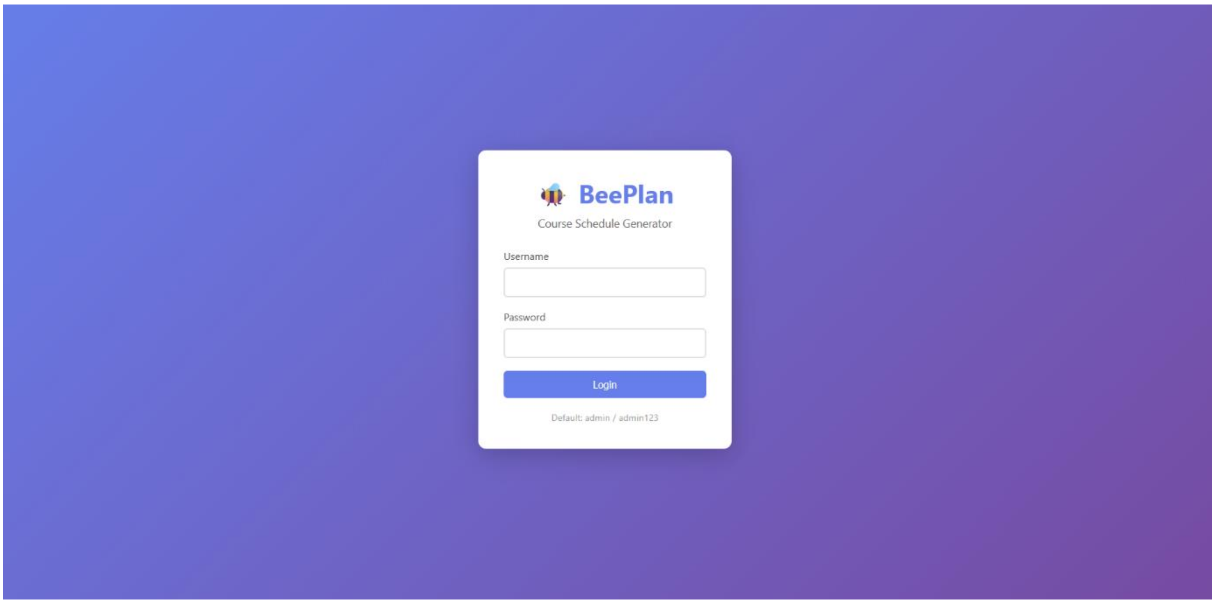
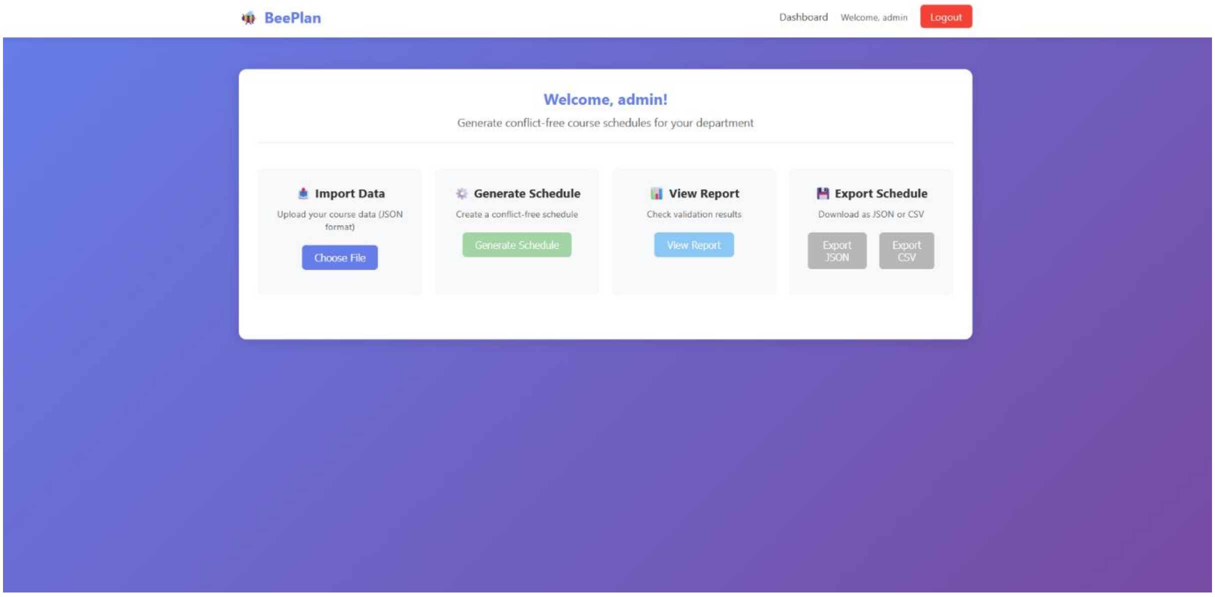


Figure 4: New version of BeePlan Modern Login Screen: A secure, clean, and professional authentication interface that ensures controlled access



Welcome, admin!

Generate conflict-free course schedules for your department

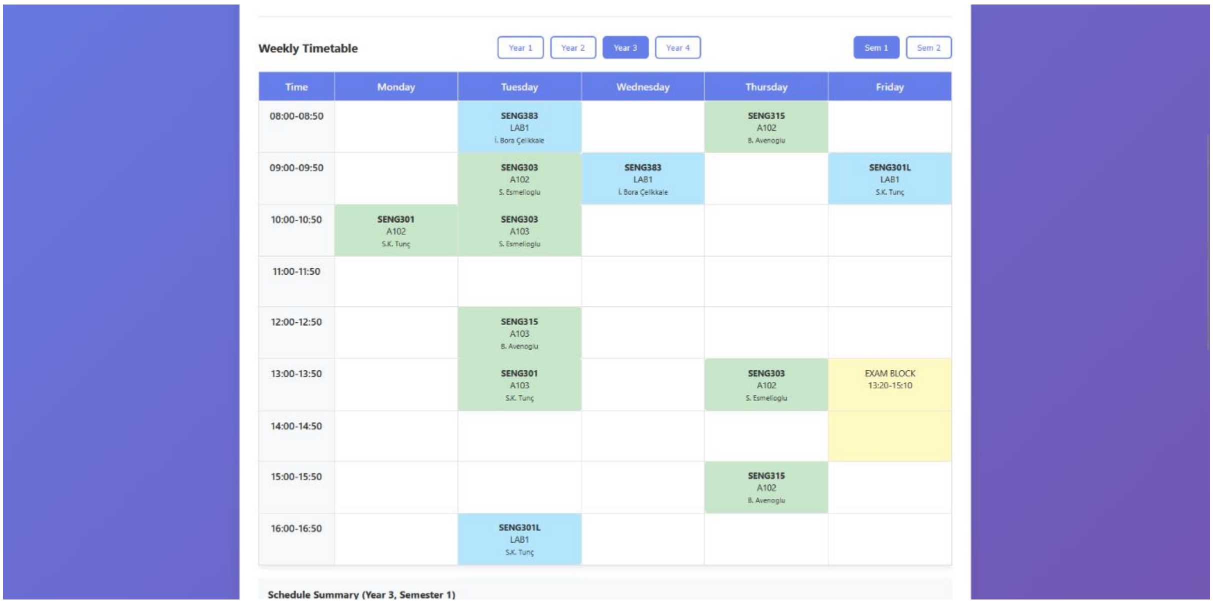
Import Data
Upload your course data (JSON format)

Generate Schedule
Create a conflict-free schedule

View Report
Check validation results

Export Schedule
Download as JSON or CSV

Figure 5: New version of BeePlan Centralized Admin Dashboard: A user-friendly control panel that manages all complex operations—such as Data Import, AI Scheduling, and Exporting—from a single hub

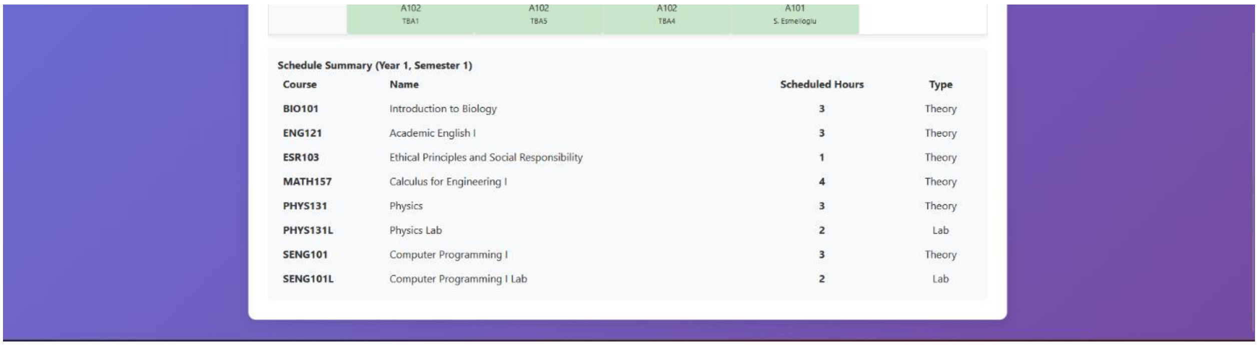


Weekly Timetable

Time	Monday	Tuesday	Wednesday	Thursday	Friday
08:00-08:50		SENG383 LAB1 İ. Bora Çelikkale		SENG315 A102 B. Avenoglu	
09:00-09:50		SENG303 A102 S. Esmelioglu	SENG383 LAB1 İ. Bora Çelikkale		SENG301L LAB1 S.K. Tunç
10:00-10:50	SENG301 A102 S.K. Tunç	SENG303 A103 S. Esmelioglu			
11:00-11:50					
12:00-12:50		SENG315 A103 B. Avenoglu			
13:00-13:50		SENG301 A103 S.K. Tunç		SENG303 A102 S. Esmelioglu	EXAM BLOCK 13:20-15:10
14:00-14:50					
15:00-15:50				SENG315 A102 B. Avenoglu	
16:00-16:50		SENG301L LAB1 S.K. Tunç			

Schedule Summary (Year 3, Semester 1)

Figure 6: New version of BeePlan Visualized Timetable: An improved weekly view where complex schedule data is made instantly understandable using color codes

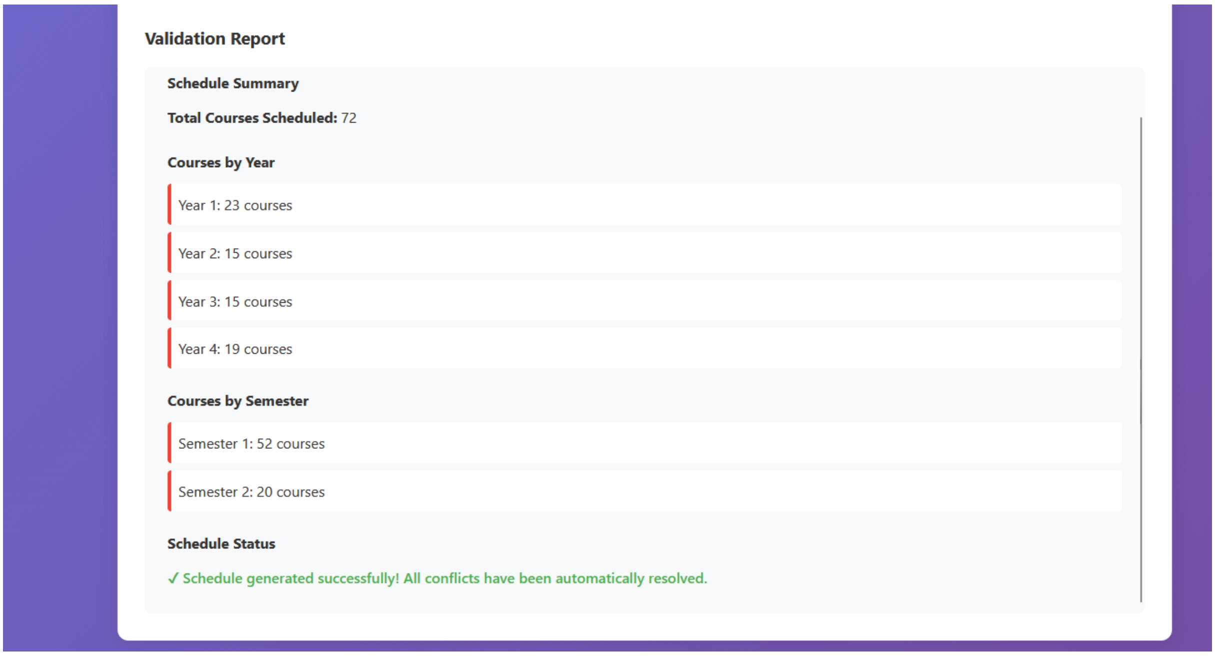


	A102 TBA1	A102 TBA5	A102 TBA4	A101 S. Esmelioglu

Schedule Summary (Year 1, Semester 1)

Course	Name	Scheduled Hours	Type
BIO101	Introduction to Biology	3	Theory
ENG121	Academic English I	3	Theory
ESR103	Ethical Principles and Social Responsibility	1	Theory
MATH157	Calculus for Engineering I	4	Theory
PHYS131	Physics	3	Theory
PHYS131L	Physics Lab	2	Lab
SENG101	Computer Programming I	3	Theory
SENG101L	Computer Programming I Lab	2	Lab

Figure 7: New version of BeePlan Detailed Course Verification: (Image 6) A tabular summary displayed below the visual calendar. It serves as a verification tool, allowing administrators to confirm that specific courses are scheduled for the correct number of hours and correctly classified as 'Theory' or 'Lab'.



Validation Report

Schedule Summary

Total Courses Scheduled: 72

Courses by Year

Year 1: 23 courses

Year 2: 15 courses

Year 3: 15 courses

Year 4: 19 courses

Courses by Semester

Semester 1: 52 courses

Semester 2: 20 courses

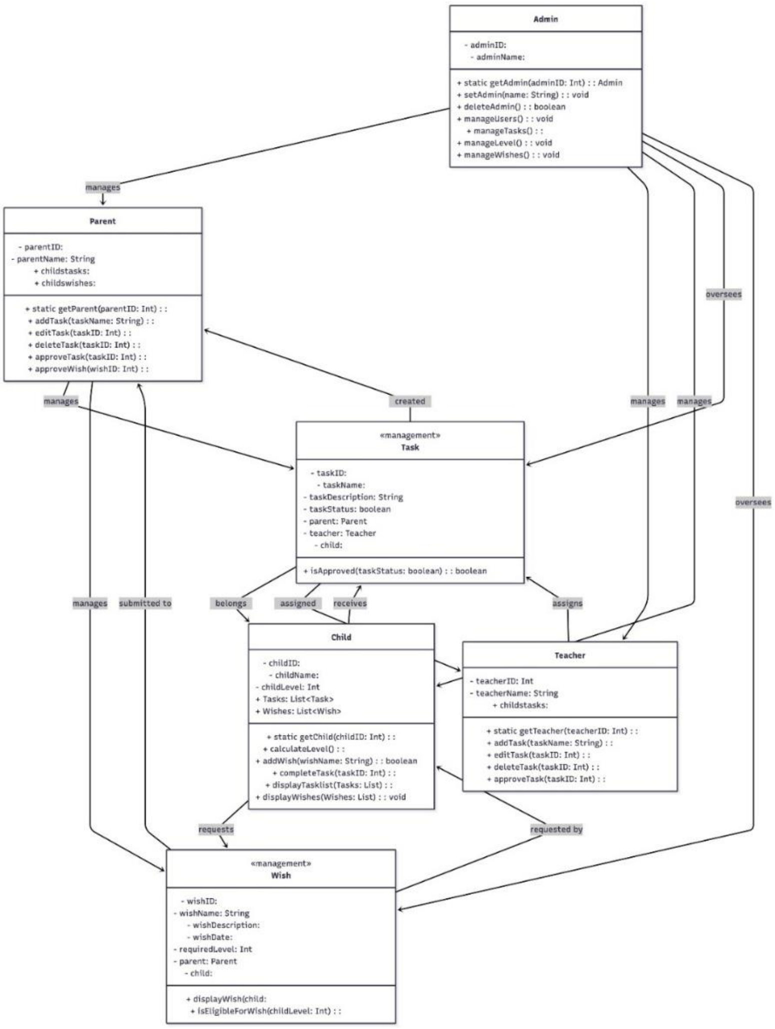
Schedule Status

✓ Schedule generated successfully! All conflicts have been automatically resolved.

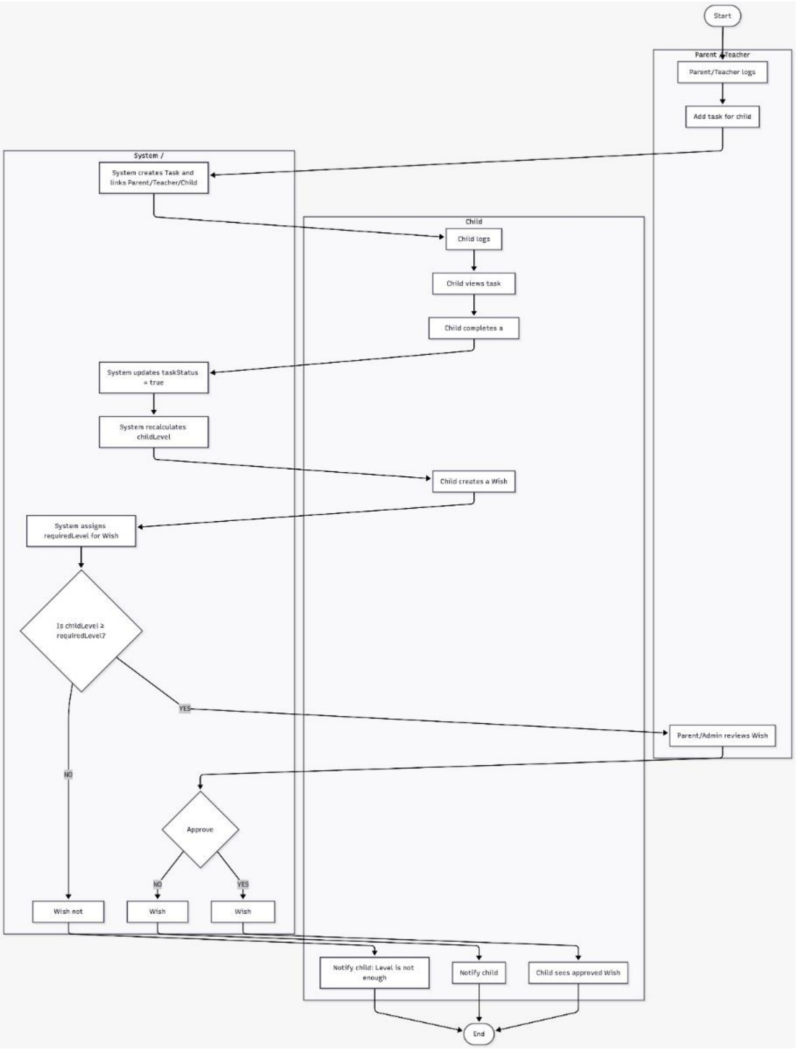
Figure 8: New version of BeePlan Clear Status Reporting: An enhanced validation screen that provides immediate and transparent feedback regarding the success of the algorithm and the distribution of courses

2.2. KidTask (Java/Spring) Design

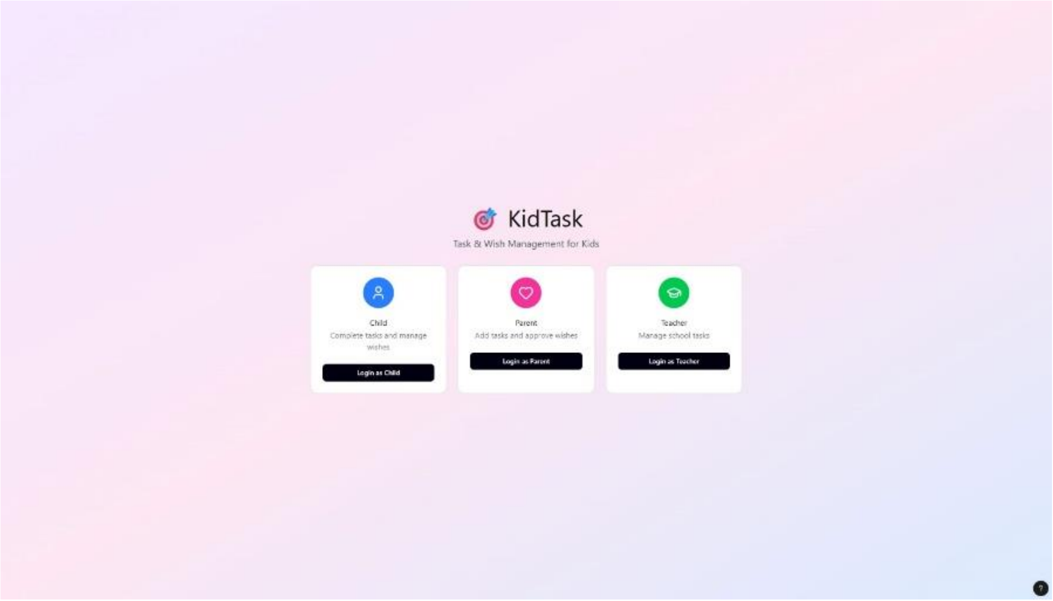
- **Class Diagram:**
  - **Classes:** Task (status, reward), Wish (cost, approval), User (budget, level), Controller (API Endpoints).


Admin
- adminID:
- adminName:
+ static getAdmin(adminID: Int) : : Admin
+ setAdmin(name: String) : : void
+ deleteAdmin() : : boolean
+ manageUsers() : : void
+ manageTasks() : :
+ manageLevel() : : void
+ manageWishes() : : void
manages
Parent
- parentID:
- parentName: String
+ childstasks:
+ childswishes:
+ static getParent(parentID: Int) : :
+ addTask(taskName: String) : :
+ editTask(taskID: Int) : :
+ deleteTask(taskID: Int) : :
+ approveTask(taskID: Int) : :
+ approveWish(wishID: Int) : :
oversees
manages
created
manages
manages
«management»
Task
- taskID:
- taskName:
- taskDescription: String
- taskStatus: boolean
- parent: Parent
- teacher: Teacher
- child:
+ isApproved(taskStatus: boolean) : : boolean
oversees
manages
submitted to
belongs
assigned
receives
assigns
Child
- childID:
- childName:
- childLevel: Int
+ Tasks: List<Task>
+ Wishes: List<Wish>
+ static getChild(childID: Int) : :
+ calculateLevel() : :
+ addWish(wishName: String) : : boolean
+ completeTask(taskID: Int) : :
+ displayTasklist(Tasks: List) : :
+ displayWishes(Wishes: List) : : void
Teacher
- teacherID: Int
- teacherName: String
+ childstasks:
+ static getTeacher(teacherID: Int) : :
+ addTask(taskName: String) : :
+ editTask(taskID: Int) : :
+ deleteTask(taskID: Int) : :
+ approveTask(taskID: Int) : :
requests
requested by
«management»
Wish
- wishID:
- wishName: String
- wishDescription:
- wishDate:
- requiredLevel: Int
- parent: Parent
- child:
+ displayWish(child:
+ isEligibleForWish(childLevel: Int) : :

- **Activity Diagram:**
  - **Flow:** Parent Adds Task -> Kid Completes Task -> Parent Approves -> **Coins Added** -> Kid Requests Wish -> Parent/System Approves -> **Coins Deducted**.


Start
Parent / Teacher
Parent/Teacher logs
Add task for child
System /
System creates Task and links Parent/Teacher/Child
Child
Child logs
Child views task
Child completes a
System updates taskStatus = true
System recalculates childLevel
Child creates a Wish
System assigns requiredLevel for Wish
Is childLevel ≥ requiredLevel?
YES
NO
Parent/Admin reviews Wish
Approve
NO
YES
Wish not
Wish
Wish
Notify child: Level is not enough
Notify child
Child sees approved Wish
End

- **GUI Screenshots:**
  - **Left:** Initial Desktop App Concept.
  - **Right:** Final Web Interface (Tasks/Wishes Tabs).


KidTask

Task & Wish Management for Kids

Child

Complete tasks and manage wishes

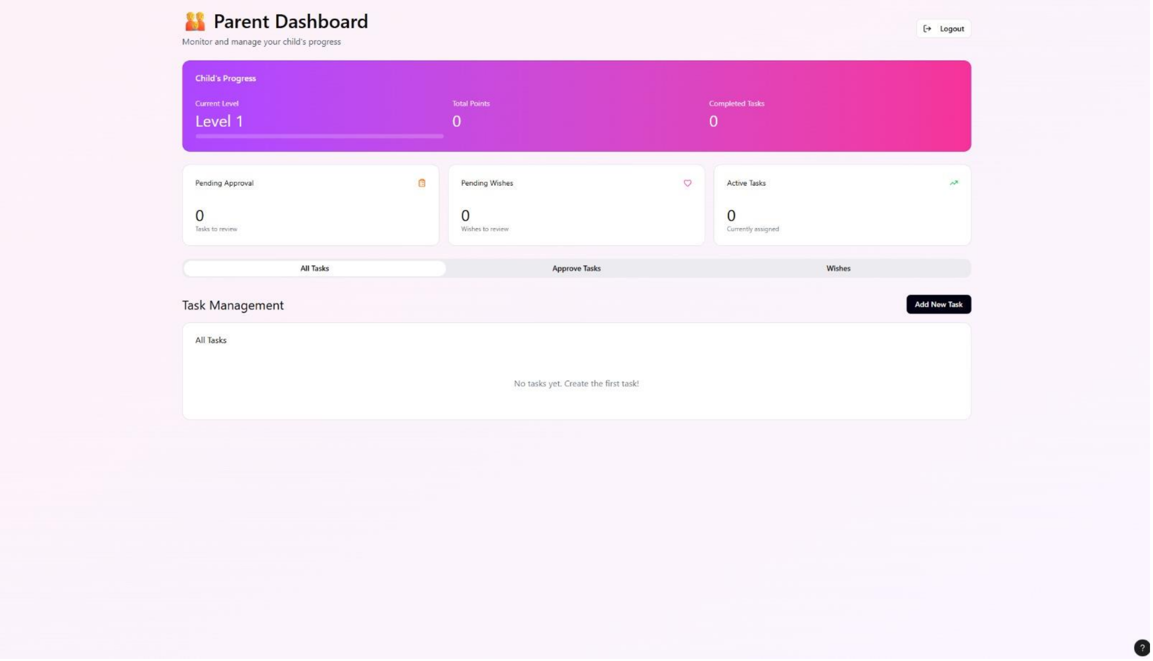
Parent

Add tasks and approve wishes

Teacher

Manage school tasks

Figure 9: A clean and simple landing screen offering three user roles — Child, Parent, and Teacher. Each role has its own card with a short description and a clear login button. The pastel background and centered layout create a modern, user-friendly first impression.



Parent Dashboard

Monitor and manage your child's progress

Child's Progress

Current Level: Level 1

Total Points: 0

Completed Tasks: 0

Pending Approval: 0 Tasks to review

Pending Wishes: 0 Wishes to review

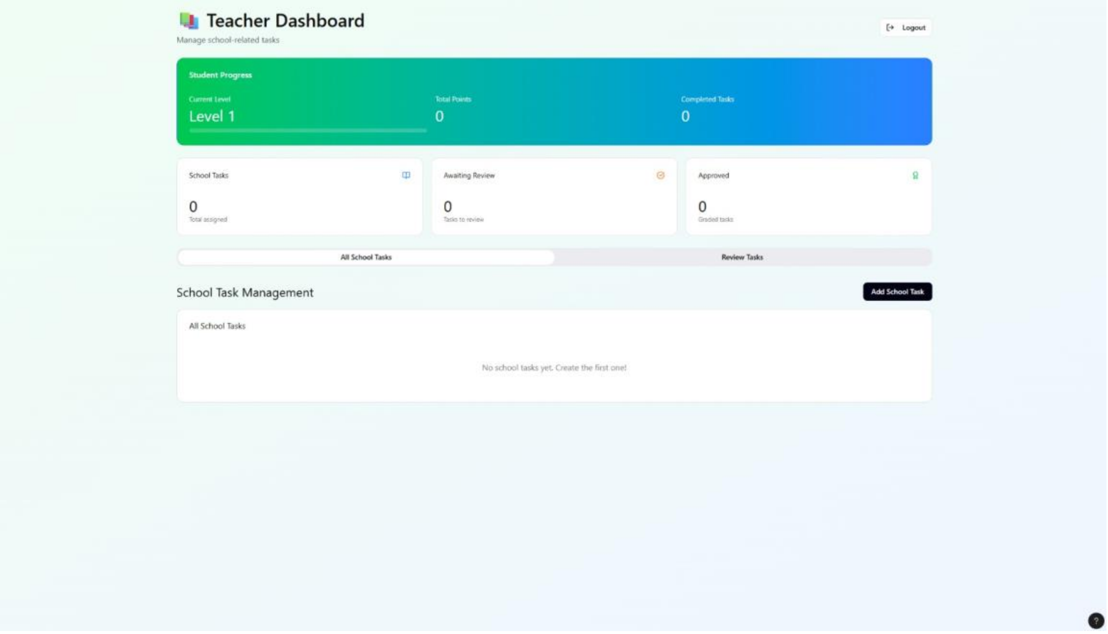
Active Tasks: 0 Currently assigned

Task Management

All Tasks

No tasks yet. Create the first task!

Figure 10: A simple dashboard where parents can track their child’s level, points, and completed tasks. They can review pending tasks, check wishes, and assign new tasks easily. The layout focuses on quick monitoring and smooth task approval.



Teacher Dashboard

Manage school-related tasks

Student Progress

Current Level: Level 1

Total Points: 0

Completed Tasks: 0

School Tasks: 0 Total assigned

Awaiting Review: 0 Tasks to review

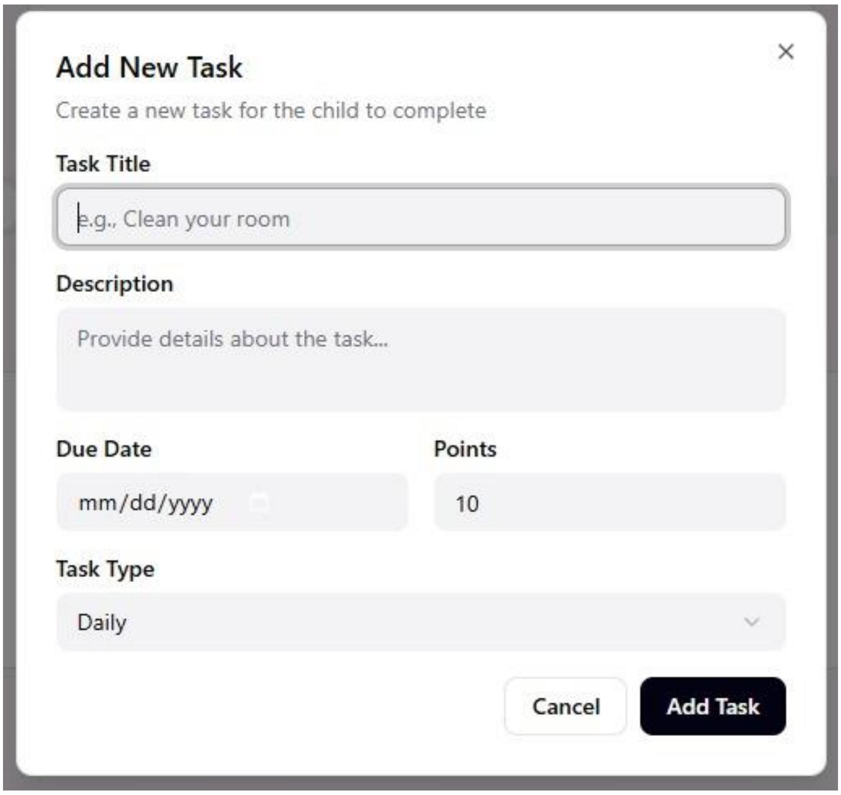
Approved: 0 Graded tasks

School Task Management

All School Tasks

No school tasks yet. Create the first one!

Figure 11: A clean dashboard showing student progress, task statistics, and management options. Teachers can view assigned, pending, and approved tasks, and easily create new school tasks. The layout is structured for quick monitoring and simple task management



Add New Task

Create a new task for the child to complete

Task Title

e.g., Clean your room

Description

Provide details about the task...

Due Date

mm/dd/yyyy

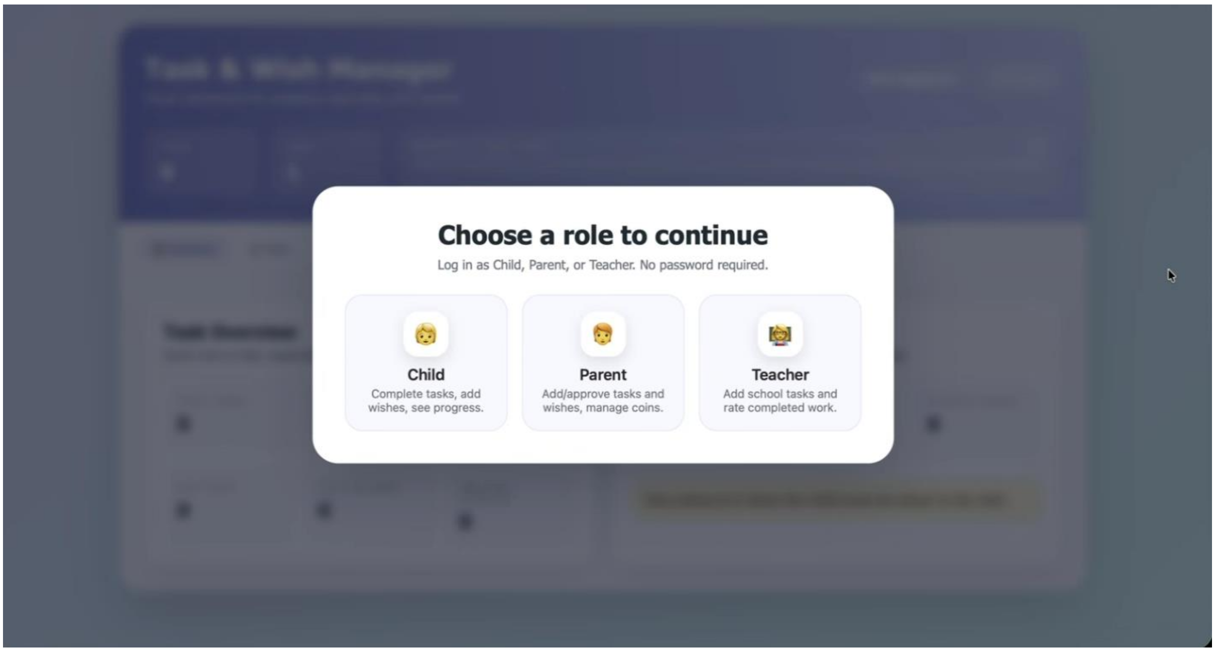
Points

10

Task Type

Daily

Figure 12: A clean modal where parents or teachers can create a new task by entering its title, description, due date, points, and type. The layout is simple and focused, making task creation quick and intuitive.



Choose a role to continue

Log in as Child, Parent, or Teacher. No password required.

Child

Complete tasks, add wishes, see progress.

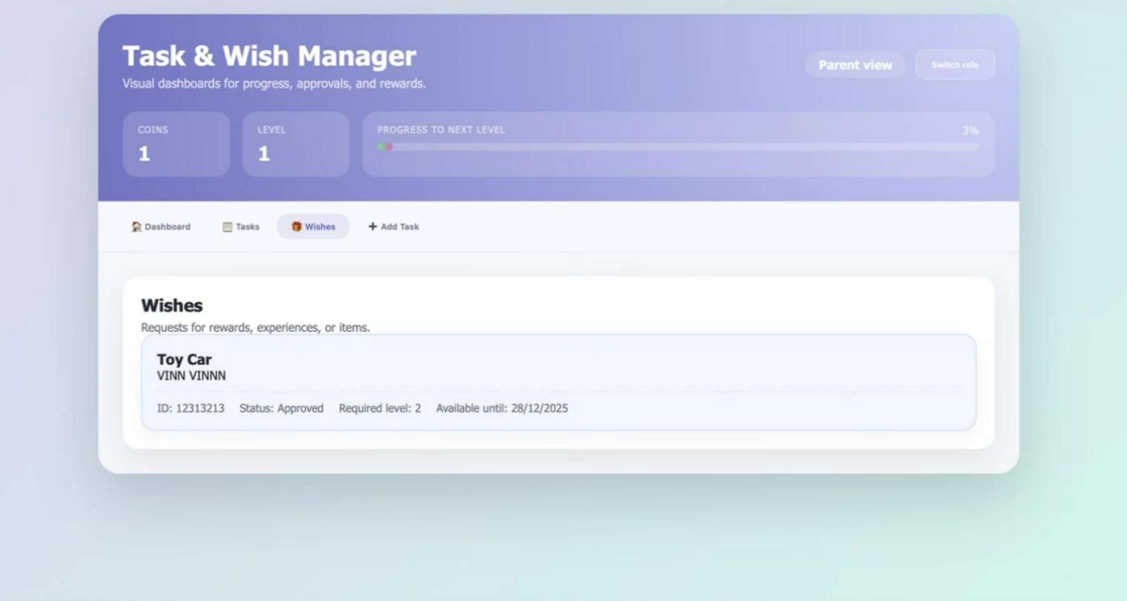
Parent

Add/approve tasks and wishes, manage coins.

Teacher

Add school tasks and rate completed work.

*Figure 13: New version of Kidtask Role-Based Access***:** *A simplified, user-friendly entry screen that allows quick switching between Child, Parent, and Teacher roles.*



Task & Wish Manager

Visual dashboards for progress, approvals, and rewards.

COINS 1

LEVEL 1

PROGRESS TO NEXT LEVEL 3%

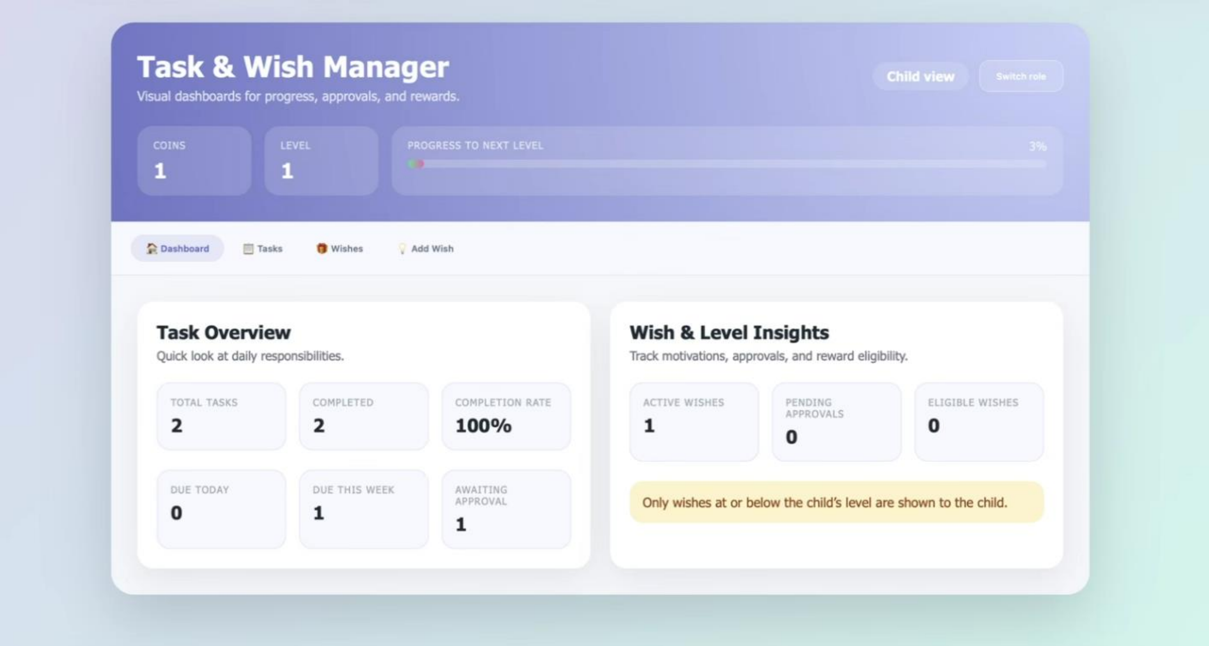
Wishes

Requests for rewards, experiences, or items.

Toy Car
VINN VINNN

ID: 12313213 Status: Approved Required level: 2 Available until: 28/12/2025

Figure 14: New version of Kidtask Gamified Dashboard: The main control center featuring real-time progress tracking, including a "Level" system, "Coin" balance, and a visual progress bar to motivate the child.



Task & Wish Manager

Visual dashboards for progress, approvals, and rewards.

COINS 1

LEVEL 1

PROGRESS TO NEXT LEVEL 3%

Task Overview

Quick look at daily responsibilities.

TOTAL TASKS 2

COMPLETED 2

COMPLETION RATE 100%

DUE TODAY 0

DUE THIS WEEK 1

AWAITING APPROVAL 1

Wish & Level Insights

Track motivations, approvals, and reward eligibility.

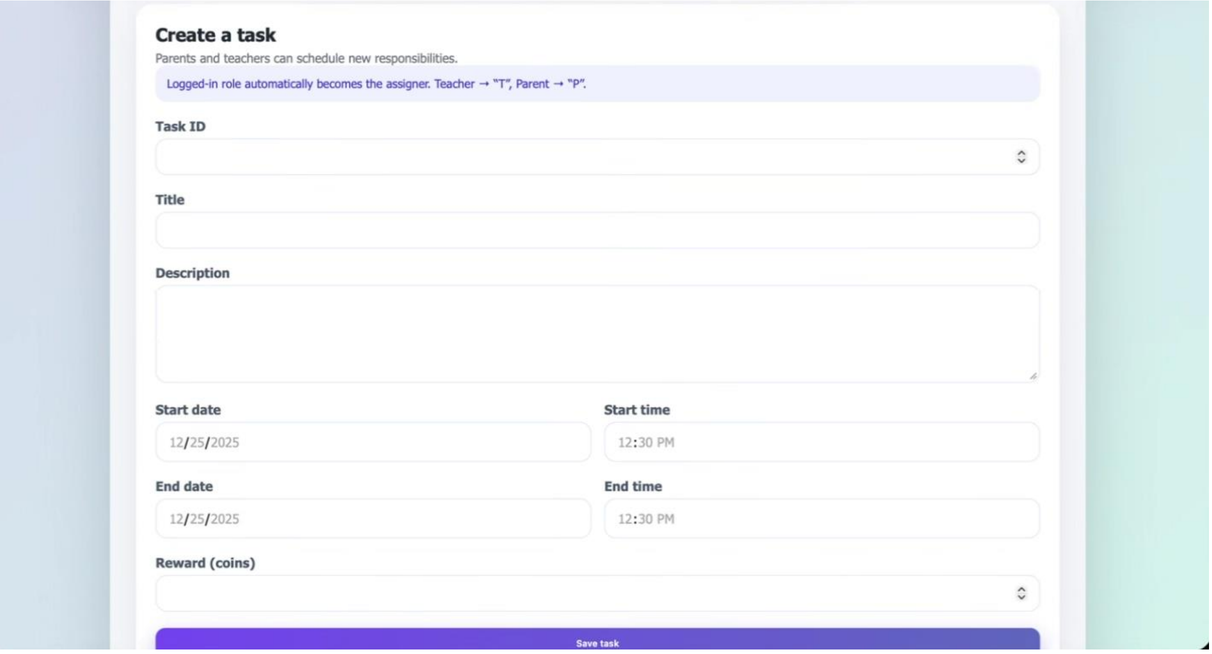
ACTIVE WISHES 1

PENDING APPROVALS 0

ELIGIBLE WISHES 0

Only wishes at or below the child's level are shown to the child.

Figure 15: New version of Kidtask Reward Management: The "Wishes" tab, where children can track the status of their requested rewards (e.g., "Toy Car") and see if the parent has approved them.



Create a task

Parents and teachers can schedule new responsibilities.

Logged-in role automatically becomes the assigner. Teacher → "T", Parent → "P".

Task ID

Title

Description

Start date: 12/25/2025

Start time: 12:30 PM

End date: 12/25/2025

End time: 12:30 PM

Reward (coins)

Figure 16: New version of Kidtask Task Assignment Module: A detailed form allowing parents or teachers to create specific tasks with defined start/end dates and assigned "Coin Rewards" to incentivize completion.

3. AI Usage Analysis (AI Usage & Prompts)

Detailed analysis of the "Prompt > Output > Revision" cycle used by both developers (Zeynep & Mert) during the project.

3.1. Tools Performance Analysis

This section evaluates the accuracy and usability of the AI and diagramming tools utilized during the development process.

1. ChatGPT (GPT-5.1) *A large-scale language model used to generate, correct, and refine UML diagrams and system logic.*

- **Accuracy:**
  - Provides highly accurate code generation for **Mermaid.js** diagrams based on natural-language descriptions.
  - Minimizes syntax errors by automatically adjusting structures (e.g., node names, decision shapes, class definitions).
  - Ensures consistency across diagrams (Class → Activity → Sequence).
- **Ease of Use:**

    - Converts complex system descriptions into working diagram code with minimal user input.
    - Allows iterative refinement (e.g., “fix layout”, “make it top-down”, “add swimlanes”).
    - Reduces manual diagramming effort by automating layout, structure, and syntax.

2. Mermaid.js *A lightweight diagramming tool that renders diagrams using text-based syntax inside Markdown.*

- **Accuracy:**
    - Renders Class and Activity diagrams reliably once the syntax is correct.
    - Provides standardized diagram shapes (classes, flows, decisions), ensuring structural clarity.
    - Integrates stably with GPT outputs.
- **Ease of Use:**
    - Very simple Markdown-like syntax; easy to edit and customize.
    - Fast rendering inside GPT, documentation tools, and code platforms.
    - No external software or GUI needed—diagrams are generated directly from plain text.

3. Markdown Rendering Environment *Used as a host for Mermaid code blocks (GitHub/VS Code).*

- **Accuracy:**
    - Ensures consistent rendering of Mermaid features supported by the environment.
    - Displays diagrams exactly as defined in the code, enabling precise verification.
- **Ease of Use:**
    - Simple copy–paste workflow.
    - Supports inline editing and versioning (crucial for project documentation).
    - Works in ChatGPT, GitHub, VS Code, and Notion without complex configuration.

3.2. Prompt & Revision Cycle

Project & Phase	Task	Tool Used	Prompt (Command)	AI Output Analysis	Human Revision (Intervention)
BeePlan (Design)	GUI Design	Canva/Figma	"Create a user dashboard layout for a scheduling app with a modern look."	The suggested color palette was inconsistent (Neon colors).	We manually corrected the palette to professional academic colors (Green/Blue/White).
BeePlan (Coding)	Scheduling Algorithm	Gemini	"Write a Python function using backtracking to solve the scheduling constraint problem."	The code worked but ignored the 'Friday Exam Block' constraint.	Added a manual if condition to block Friday 13:20-15:10 slots.
KidTask (Coding)	Spring Controller	Claude	"Write a Spring Boot Controller for Task management with GET/POST endpoints."	The code worked but generated a CORS error when connecting to the frontend.	Added @CrossOrigin("*") annotation to the controller class manually.
Both (Reporting)	Documentation	Gemini	"Summarize the testing phase and write a conclusion for the report."	The AI used overly complex technical jargon suitable for a Ph.D. thesis.	We rewrote the summary in simpler, clearer language suitable for the project scope.

4. V&V Test Reports (Verification & Validation)

This section documents the testing process, bug fixes, and peer reviews.

4.1. Test Case Table

Sample test cases executed for both systems.

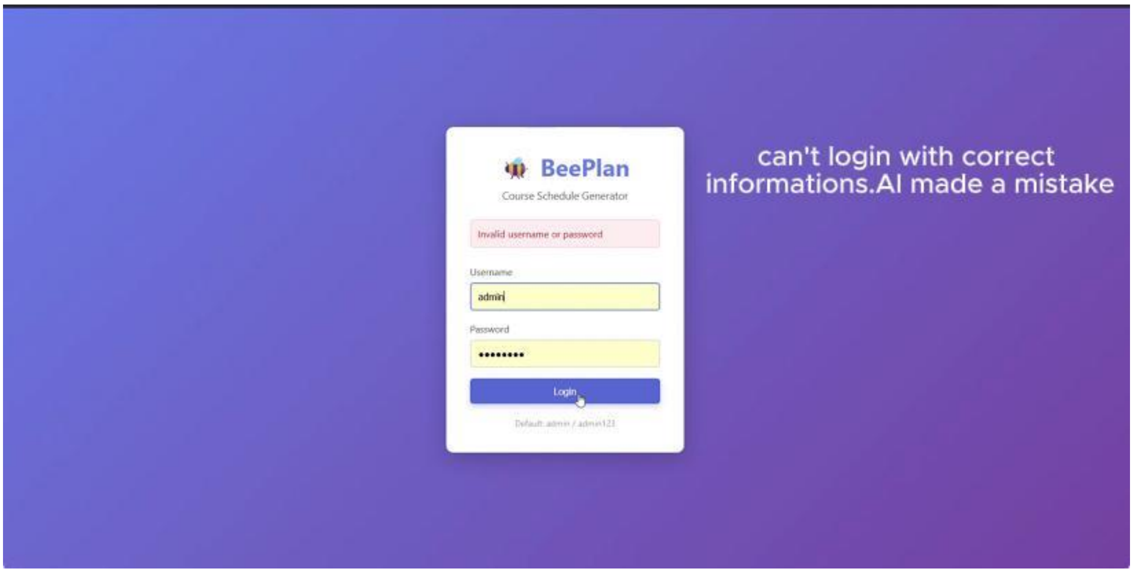
Test ID	Project	Scenario	Expected Result	Actual Result	Status
TC-01	BeePlan	Admin Uploads Invalid JSON	System shows error message "Invalid Format".	System crashed (500 Error).	**Fixed**
TC-02	BeePlan	Generate Schedule (Clean)	No conflicts between instructors.	Schedule generated successfully.	**Pass**
TC-03	KidTask	Complete Task	Coin balance increases by task value.	Coin balance increased correctly.	**Pass**
TC-04	KidTask	Request Wish (Insufficient Funds)	System rejects request with "Not enough coins".	System allowed negative balance.	**Fixed**

4.2. AI Tutor Bug Solutions

Critical bugs encountered during development and solved using the "Human-in-the-Loop" methodology.

Bug 1: BeePlan - Authentication Failure (Password Hashing)

- **Issue:** When tasked with creating the login function, the AI implemented a plain text password comparison (if db_password == input_password). Since the database stores passwords as secure hashes, valid login attempts always returned "Invalid username or password".
- **AI Tool:** Gemini (Backend Logic)
- **Solution (Human Intervention):** We corrected the logic by implementing the check_password_hash() function from the Werkzeug security library. This ensured that the input password was correctly hashed before comparison, fixing the authentication process.



BeePlan

Course Schedule Generator

Invalid username or password

Username: admin

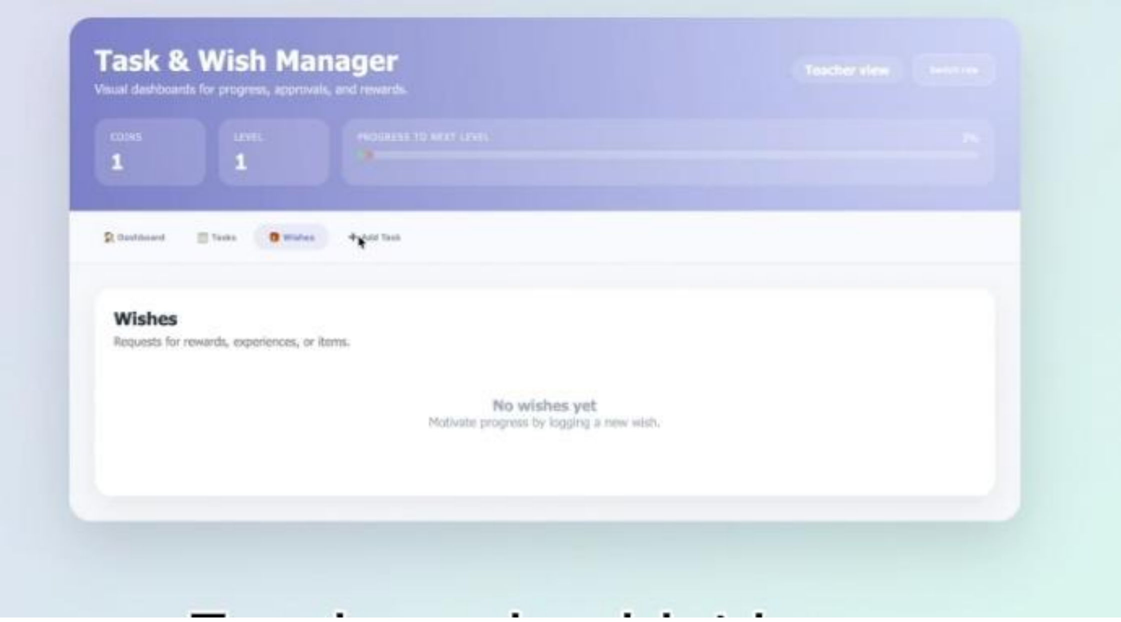
Password: ••••••••

Default: admin / admin123

can't login with correct informations.AI made a mistake

Bug 2: KidTask - Unauthorized Role Access (RBAC Flaw)

- **Issue:** The AI incorrectly granted the **"Teacher"** role access to the "Add Wish" functionality, which is logically restricted to the **"Child"** role. Teachers should only assign tasks, not create wish requests.
- **AI Tool:** Claude (Role-Based Access Control)
- **Solution (Human Intervention):** We intervened in the Controller and Frontend logic to enforce strict Role-Based Access Control (RBAC). We added a condition (if role == 'CHILD') to the "Add Wish" endpoint, effectively removing this functionality from the Teacher's interface.



Task & Wish Manager

Visual dashboards for progress, approvals, and rewards.

COINS 1

LEVEL 1

PROGRESS TO NEXT LEVEL

Wishes

Requests for rewards, experiences, or items.

No wishes yet

Motivate progress by logging a new wish.

4.3. Peer Review Findings

Cross-examination results between partners.

- **Reviewer: Mert (Student B) -> Reviewed: BeePlan (Zeynep)**
  - **Finding:** "The 'Export to CSV' button does not handle Turkish characters correctly (e.g., 'Ç' becomes '?')."
  - **Solution:** Zeynep added encoding='utf-8-sig' to the pandas export function.
- **Reviewer: Zeynep (Student A) -> Reviewed: KidTask (Mert)**
  - **Finding:** "The check_wish endpoint approves wishes even if the student's level is too low."
  - **Solution:** Mert added a Logic Gate: if (user.level >= wish.requiredLevel) before approving.