KIDTASK

Task and Wish Management Application (Java GUI Version)

Final Project Report

1. Project Identity and Repository Access

Project Name: KidTask – Task and Wish Management Application
Developer: Dilan yardım – Role: Student A
AI Tool Used: DeepSeek (AI Tutor)
Programming Language: Java

GitHub Repository:
https://github.com/dilan11233/SENG383-project

2. Final Design Artifacts

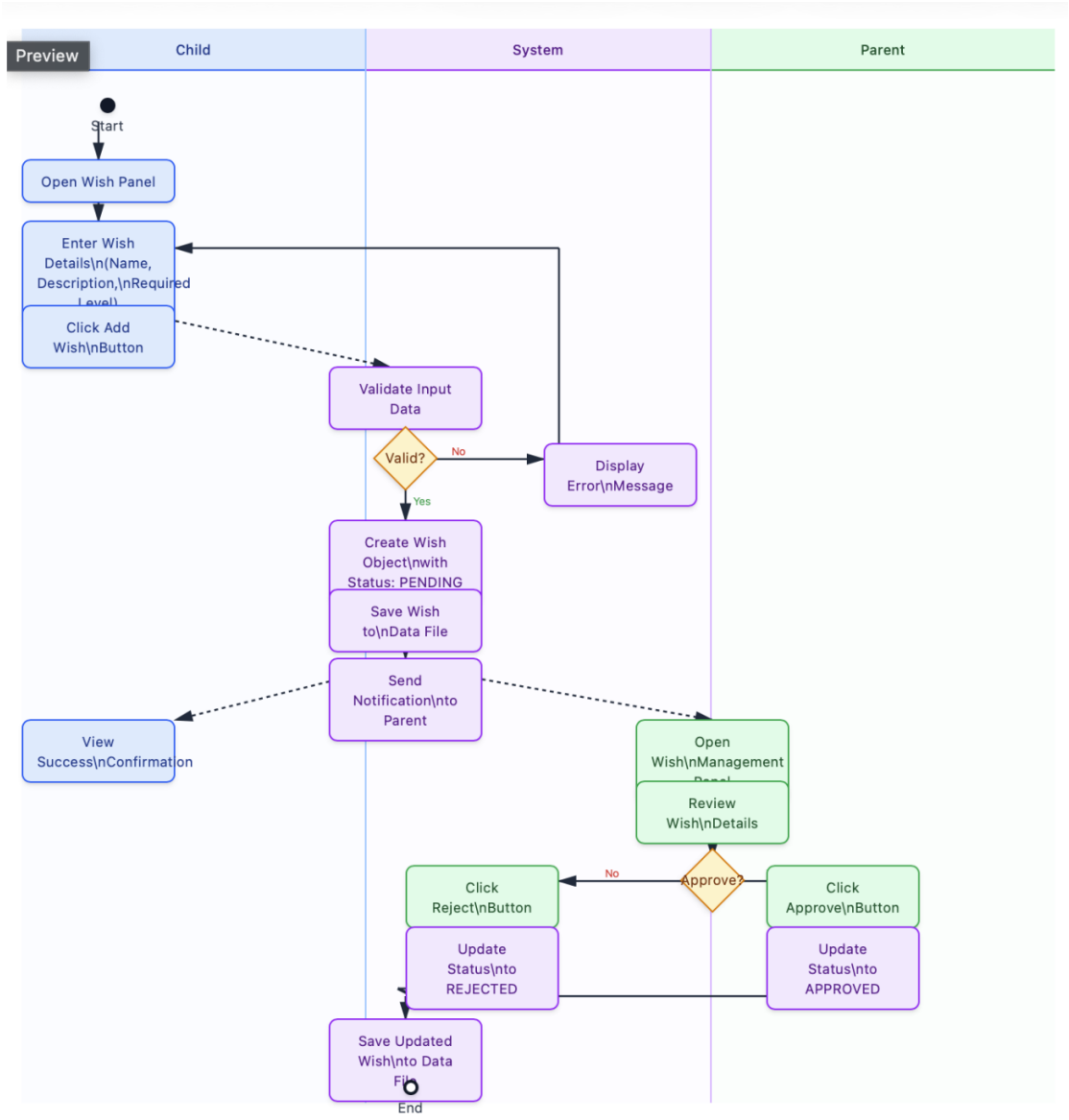
2.1 Class Diagram

The class diagram represents the final implementation of the KidTask system.
Core classes include:

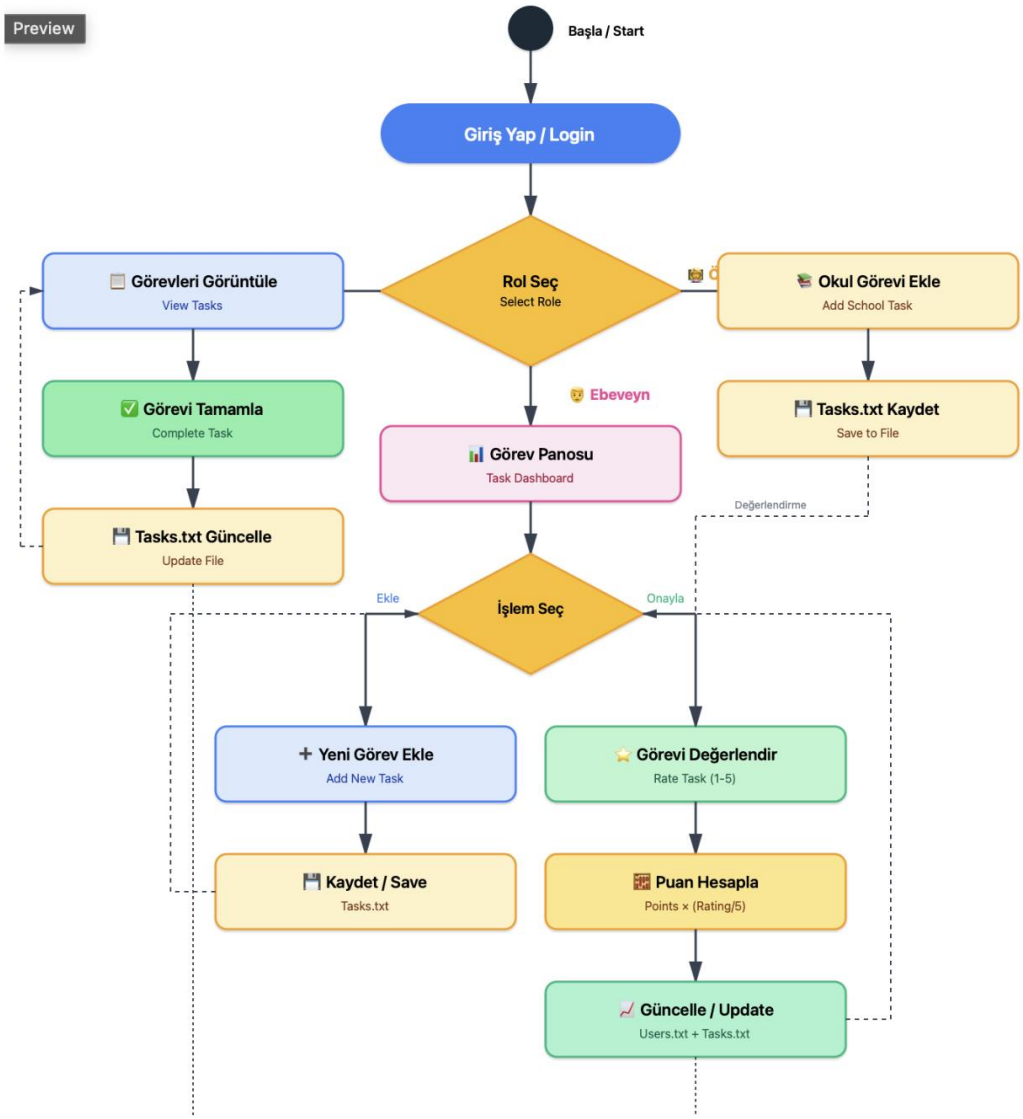
- User (Child, Parent, Teacher)
- Task
- Wish
- TaskManager
- WishManager
- PointManager
- FileManager

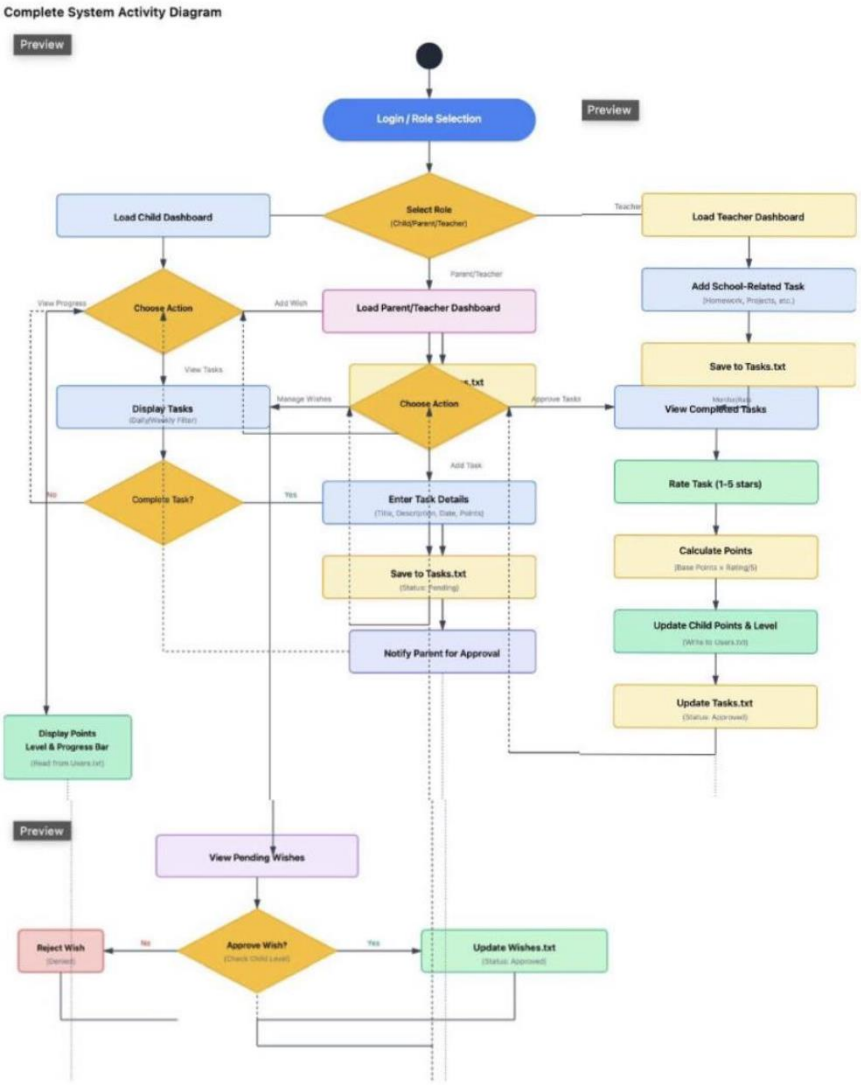
Compared to the initial design, task approval and rating logic were separated into dedicated classes to improve maintainability and support role-based authorization.

2.2 Activity Diagram (User Scenario)



Preview


Başla / Start
Giriş Yap / Login
Rol Seç
Select Role
Görevleri Görüntüle
View Tasks
Okul Görevi Ekle
Add School Task
Görevi Tamamla
Complete Task
Ebeveyn
Tasks.txt Kaydet
Save to File
Görev Panosu
Task Dashboard
Değerlendirme
Tasks.txt Güncelle
Update File
Ekle
İşlem Seç
Onayla
Yeni Görev Ekle
Add New Task
Görevi Değerlendir
Rate Task (1-5)
Kaydet / Save
Tasks.txt
Puan Hesapla
Points × (Rating/5)
Güncelle / Update
Users.txt + Tasks.txt



The activity diagram reflects the main user interaction flow demonstrated in the final presentation video:

1. User selects role (Child / Parent / Teacher)
2. User logs into the system
3. Dashboard is displayed
4. User performs role-specific actions
    a. Child: complete tasks, add wishes
    b. Parent/Teacher: approve tasks, rate performance
5. System updates points, levels, and stored data
6. Validation messages and dashboards are refreshed

This diagram accurately represents the final behavior of the application.

2.3 GUI Screenshots

The final GUI design differs significantly from the initial wireframe version.

Key improvements include:

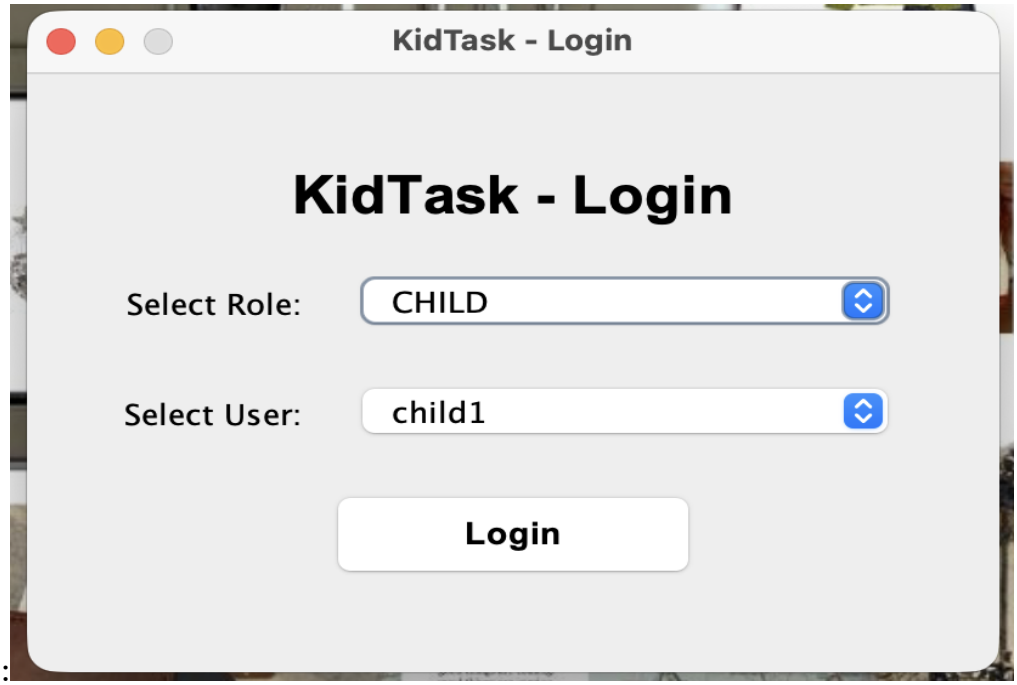
- Dedicated panels for Task Management and Wish Management
- Visual progress indicators for points and levels
- Clear approval and rejection dialogs for parents and teachers

The final interface improves usability and clarity while maintaining a child-friendly design.

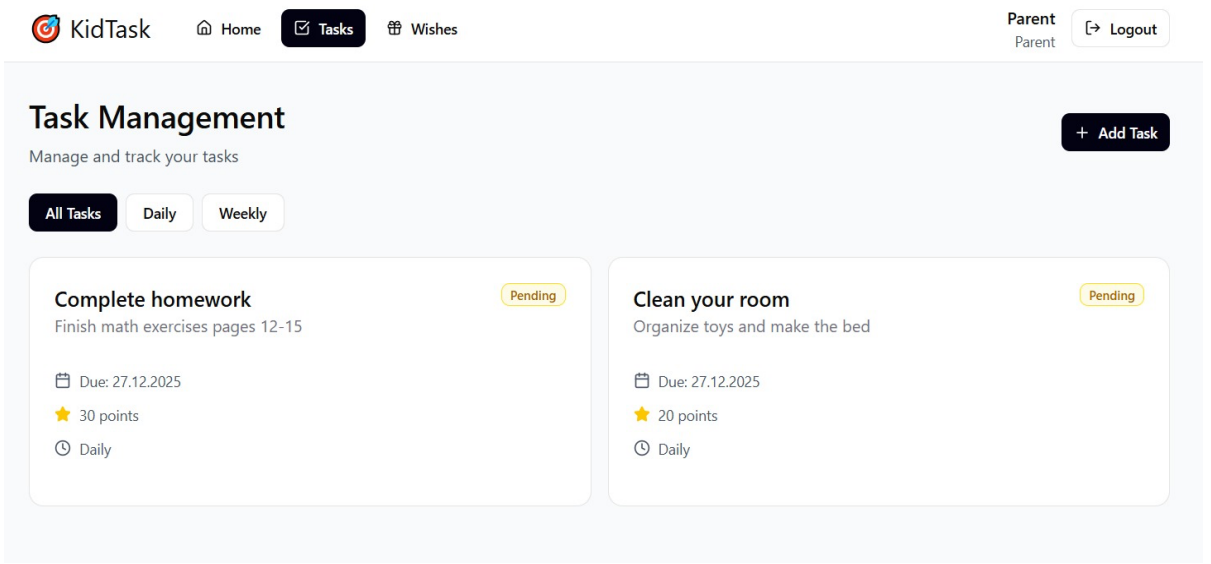
First design:

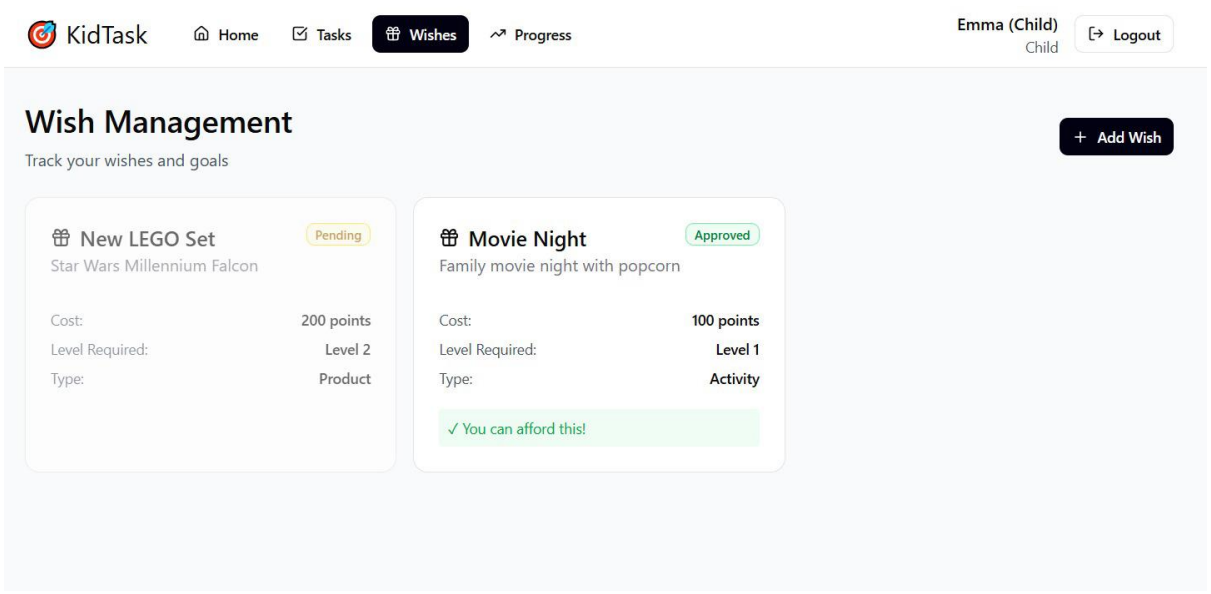
No design for login.

Final GUI:

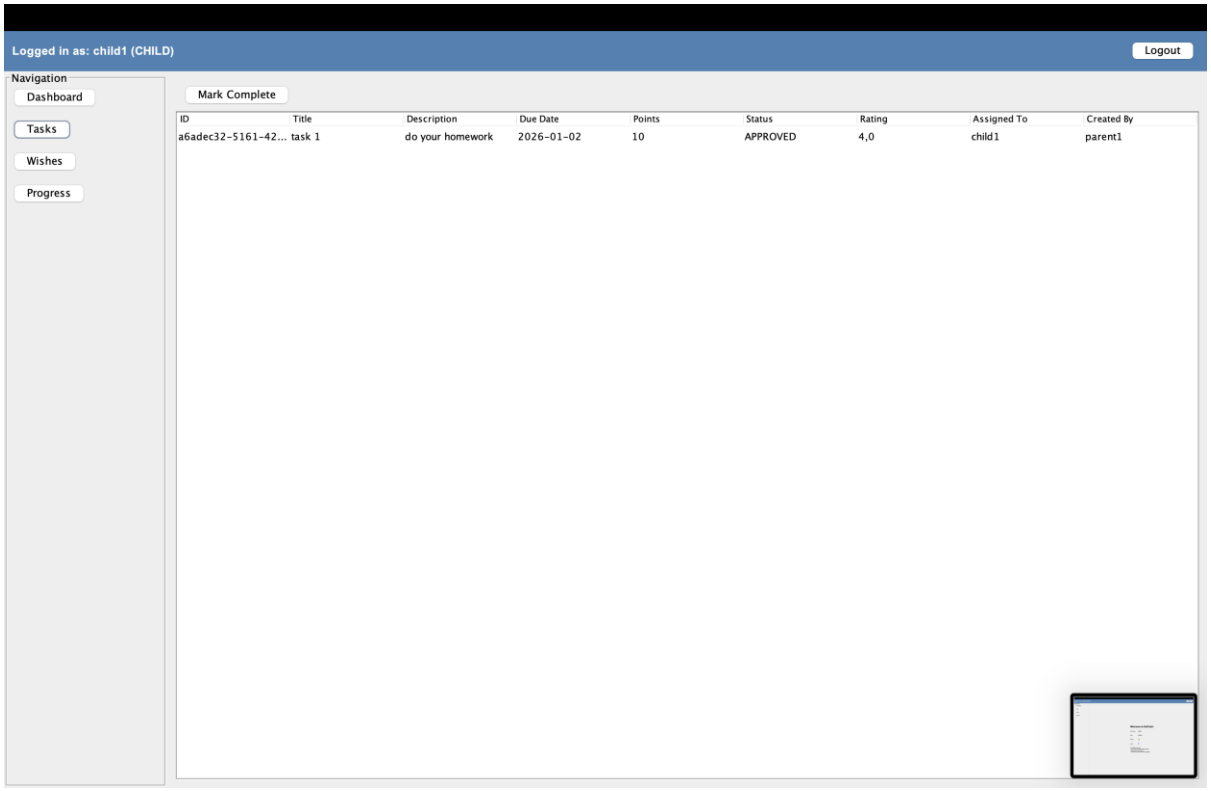
: 

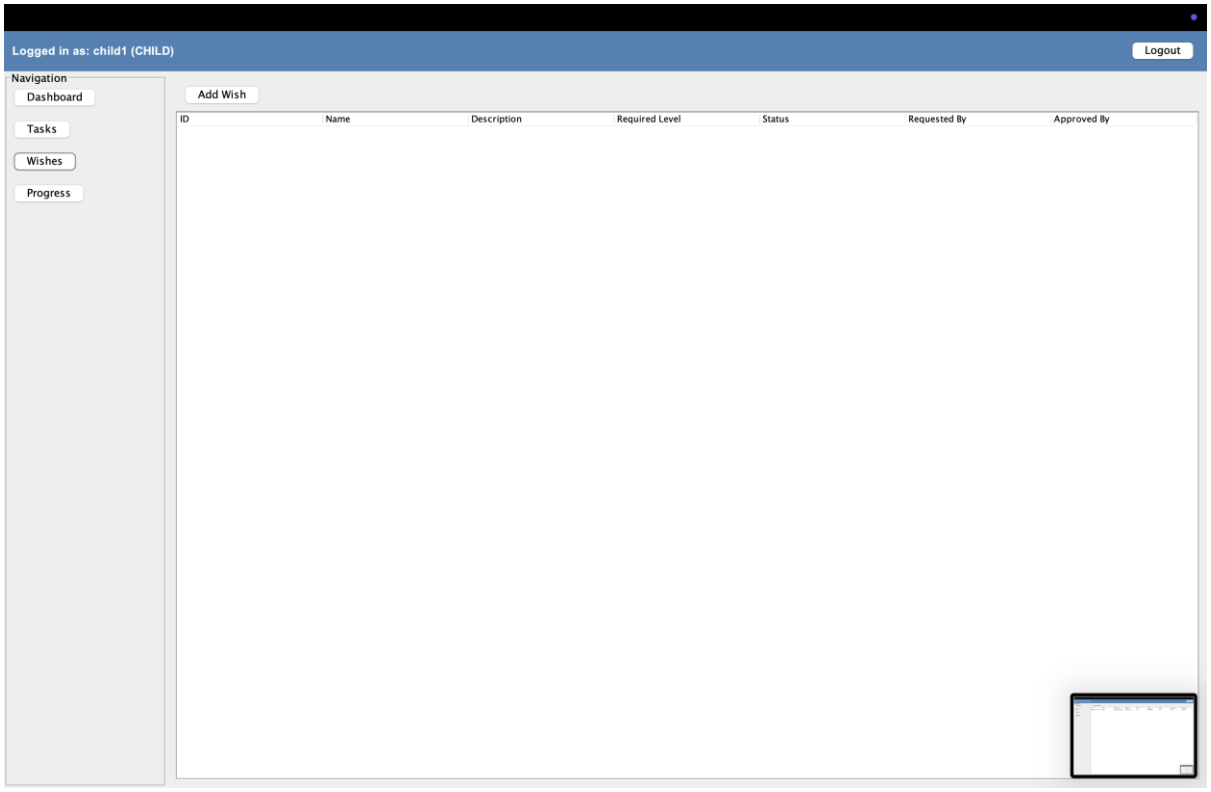
First design Task-Wish:


KidTask
Home
Tasks
Wishes
Parent
Parent
Logout
Task Management
Manage and track your tasks
Add Task
All Tasks
Daily
Weekly
Complete homework
Pending
Finish math exercises pages 12-15
Due: 27.12.2025
30 points
Daily
Clean your room
Pending
Organize toys and make the bed
Due: 27.12.2025
20 points
Daily

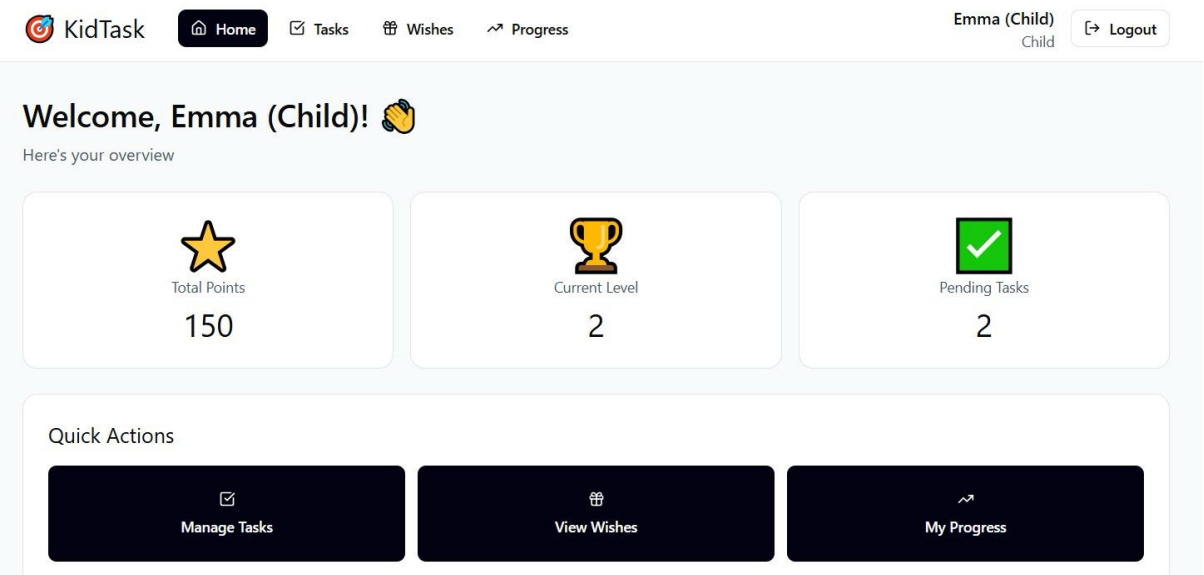

KidTask
Home
Tasks
Wishes
Progress
Emma (Child)
Child
Logout
Wish Management
Track your wishes and goals
Add Wish
New LEGO Set
Pending
Star Wars Millennium Falcon
Cost:
200 points
Level Required:
Level 2
Type:
Product
Movie Night
Approved
Family movie night with popcorn
Cost:
100 points
Level Required:
Level 1
Type:
Activity
✓ You can afford this!

Last Design:


Logged in as: child1 (CHILD)
Logout
Navigation
Dashboard
Tasks
Wishes
Progress
Mark Complete
ID
Title
Description
Due Date
Points
Status
Rating
Assigned To
Created By
a6adec32-5161-42...
task 1
do your homework
2026-01-02
10
APPROVED
4,0
child1
parent1

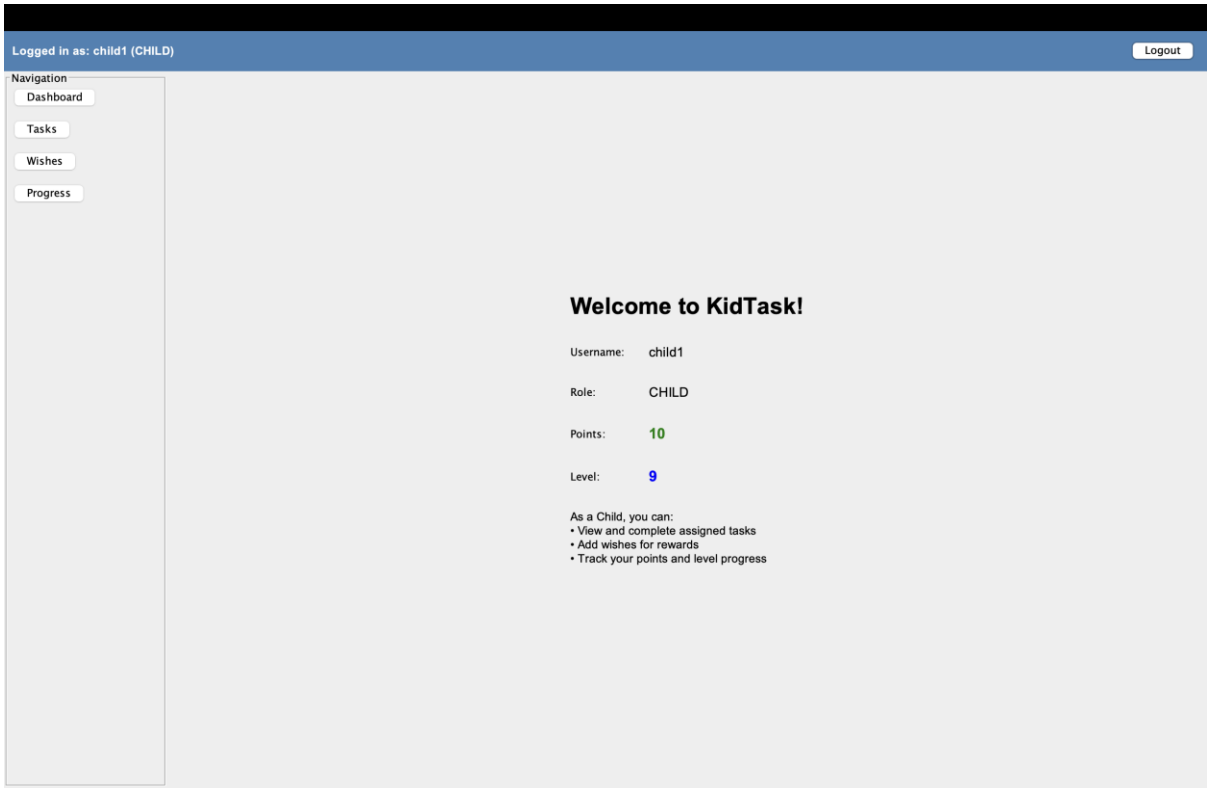

Logged in as: child1 (CHILD)
Logout
Navigation
Dashboard
Tasks
Wishes
Progress
Add Wish
ID
Name
Description
Required Level
Status
Requested By
Approved By

Dashboard first design:

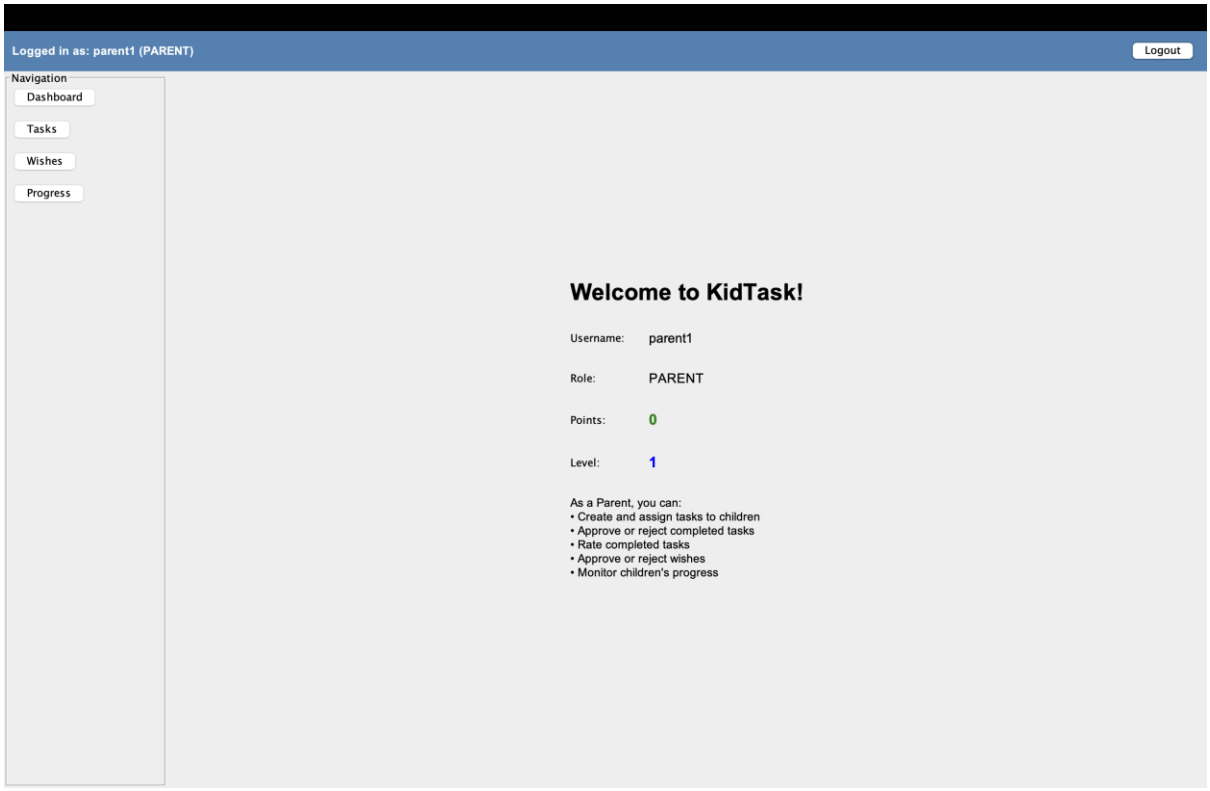


Final design:

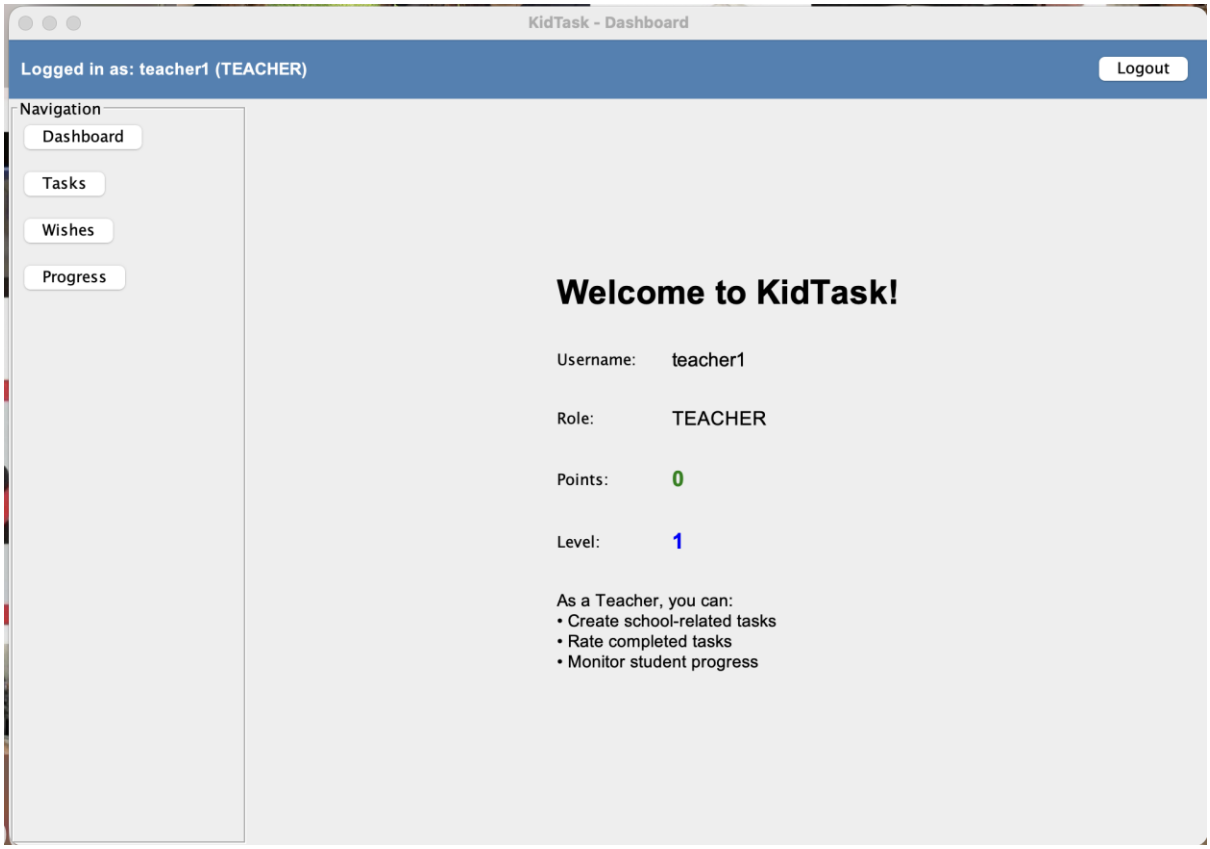
Child :



Parent:



Teacher:



3. AI Usage and Prompt Analysis

AI tools were used as **tutoring assistants**, not as direct code generators.
All AI-generated outputs were reviewed, revised, and manually integrated into the project.

AI Usage Summary Table

Stage	Prompt Summary	AI Output	Student Revision
Design	How to model task approval and roles	Suggested state-based task flow	Implemented role-based logic
Coding	How to prevent points update before approval	Recommended separating states	Added task status enum
Testing	Generate negative test cases	Provided FAIL scenarios	Adapted to project rules

AI assistance improved productivity and design clarity while keeping full control with the developer.

4. Verification and Validation (V&V)

Verification and validation activities were conducted to ensure that the KidTask system satisfies all functional and technical requirements.
Both **positive (PASS)** and **negative (FAIL)** test cases were executed.

4.1 Test Case Summary Table

Test Case	Scenario	Expected Result	Actual Result	Status
TC-01	Child completes task	Task pending approval	As expected	PASS
TC-02	Parent approves task	Points updated	As expected	PASS
TC-03	Child adds wish	Wish pending approval	As expected	PASS
TC-F01	Task without approval	Points unchanged	Correctly blocked	FAIL (Expected)

TC-F02	Wish below level	Not displayed	Correctly hidden	FAIL (Expected)

4.2 FAIL Test Case Analysis (Validation)

TC-F01: Task Completion Without Approval

Description:
A child completes a task, but the task is not approved by a parent or teacher.

Expected Result:

- Points and level should not be updated
- Task remains in “Waiting Approval” state

Result:
The system correctly prevents point updates.
This FAIL result confirms correct authorization control.

TC-F02: Wish Below Child Level

Description:
A child attempts to view a wish that requires a higher level.

Expected Result:

- Wish should not be visible

Result:
The system hides ineligible wishes correctly, validating business rule enforcement.

4.3 AI Tutor Assisted Bug Fixing

Bug 1: Points Updated Before Approval

Problem:
Points were incorrectly updated immediately after task completion.

AI Tool Used: DeepSeek

Solution:
AI suggested separating task completion and approval states.

Fix Applied:
A task status system (PENDING, COMPLETED, APPROVED) was implemented.

Outcome:
Bug resolved successfully.

Bug 2: Wishes Displayed Regardless of Level

Problem:
All wishes were visible to children regardless of their level.

AI Tool Used: DeepSeek

Solution:
AI recommended filtering wishes based on level.

Fix Applied:
Level-based filtering was added to the wish display logic.

Outcome:
Only eligible wishes are displayed.

4.4 Peer Review Findings

Peer review (Dilan/Beril) identified the following issues:

- Missing validation for empty task input fields
- Tight coupling between task completion and approval logic

These issues were resolved by:

- Adding input validation checks
- Refactoring task state management

Peer feedback improved system reliability and maintainability.

5. Conclusion

KidTask successfully meets all functional and technical requirements defined for the project.
The system provides a role-based, user-friendly GUI that allows children to manage tasks and wishes while ensuring parental and teacher control.

AI tools were used responsibly as tutoring assistants, contributing to design quality and debugging efficiency without replacing developer decision-making.

The final product demonstrates effective use of GUI programming, file-based persistence, validation techniques, and AI-assisted development practices.