SENG 383 - Software Project III raport

On the design part of this report we have used Canva and Figma to create GUI design and class-activity diagrams for kidTask and BEEplan projects. I used Canva to generate GUI design for BEEplan and my partner(Beril Aşçi) used Figma to generate diagrams for kidTask; ı used Figma to generate diagrams for kidTask and Beril used Canva to generate GUI design for kidTask.

Canva-GUI:

Canva has two versions of AI where you can create GUI: design and image.

OUTPUT QUALITY:

Design version of Canva had a better image for creating presentations but unfortunately it lacked the creativity and detailed imagery for GUI design therefore was unable to generate the wanted designs. Image version on the other hand was able to crate the overall goal image but the details were very confusing and the letters were all wrong. You need to keep making changes and update the final image which made the final image even more confusing.

USABILITY

When you look at the reaction time and AI limitations , Canva was a better choice for reaction time ;it only takes a few seconds to create the image and you can keep making changes on the product. For the prompts you need for generating image you have a limitation for 4000 characters and a limit of 50 for AI images. But after you run out of the limited AI usage you have to wait until next month to be able to use AI mode. For this reason some images had to stay as they are since ı was unable to use AI to adjust the image to corrected versions.

FIGMA-DIAGRAMS:

For kidTask i had aimed to create 5 diagrams: Login/Role Selection, Task Management, Wish Management, Point Tracking, and a final class diagram.

OUTPUT QUALITY:

Figma AI has a great understanding of what a diagram should be like and the generated product is mostly correct however on some versions figma confused what a class and activity diagram is and kept adding classes to activity diagrams and on some versions the lines and boxes were added on top each other. The output for the diagrams rarely needed correction and most of the details were correct.

USABILITY:

Just like Canva ,Figma also has a very easy to understand user interface. All you need to do is write down what you need the AI to create and wait for the image to download.

Figma gives you only 10-20 free AI usage however when I was using figma ,the app kept redoing the tasks to perfect the final product, and for that reason i couldn’t create the 5 diagrams I was aiming for. When i gave the first instruction to create the task manager activity diagram figma AI used 20 different files and for that reason my limit was over. I had to login from another e-mail to create rest of the activity diagrams even than I could only create the task and wish manager. The overall limitations and the fact that you have to wait for the next month to create other images for free makes the app very time consuming to use.

Conclusion

Overall, Canva is more efficient for rapid GUI prototyping but limited in precision, while Figma provides higher diagram accuracy but stricter AI quotas. Combining both tools allowed us to balance visual design quality and diagrammatic correctness, though AI limitations remain a practical challenge for large-scale academic projects.

CANVA-DIAGRAMS

Output Quality: Beril's version Canva produces clean, visually appealing, and high-resolution diagrams that are ideal for reports and presentations. Its alignment tools and export options ensure professional-looking results; however, the lack of UML-specific elements such as class compartments, association arrows, and automatic connectors limits the technical precision of activity and class diagrams. As a result, while the visual quality is excellent, users must rely on manual adjustments to maintain structural accuracy.

Usability: Canva's interface is highly intuitive, featuring drag-and-drop controls, real-time collaboration, and easy accessibility through a browser. It allows users to create and edit diagrams quickly without prior design experience. However, because UML logic must be built manually, it is less efficient for technical work. Overall, Canva is user-friendly and efficient for educational or visual purposes but not optimized for formal software modeling.

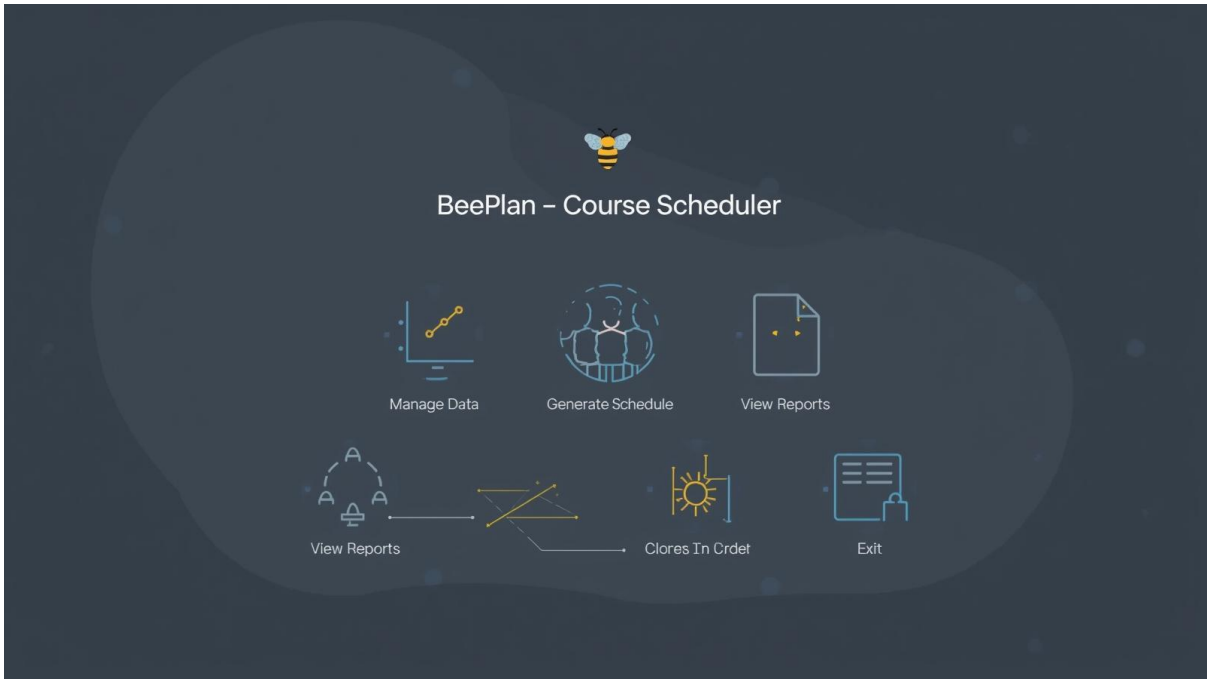
FIGMA-GUI DESIGN

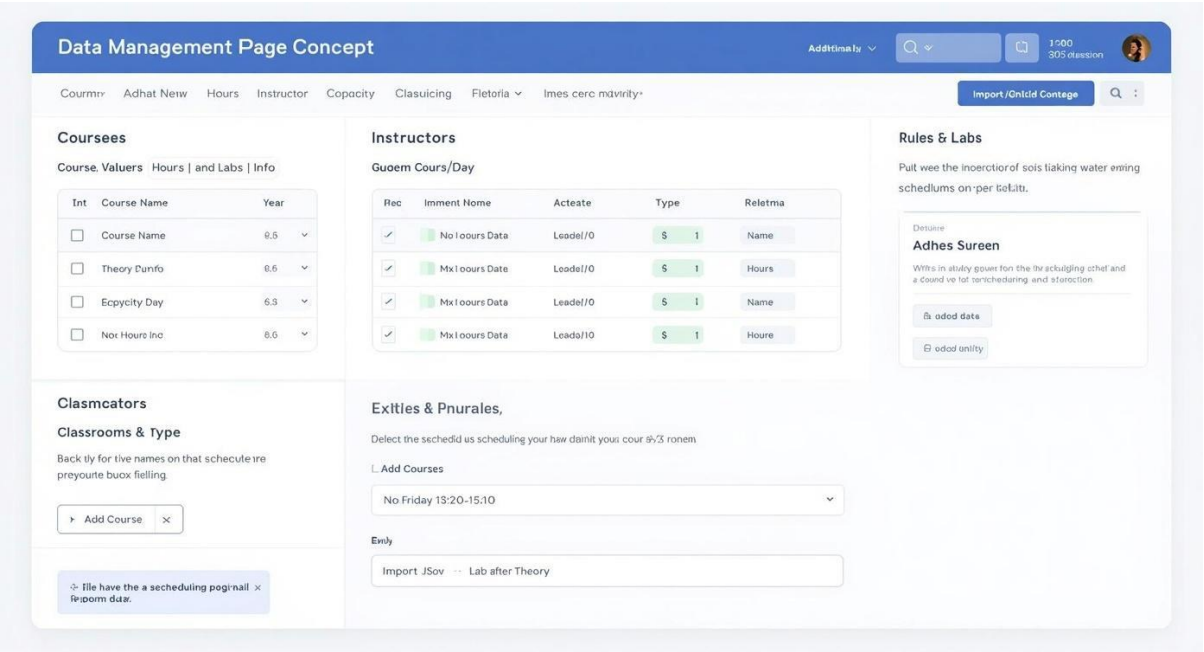
Output Quality: Figma delivers high-fidelity and professional-quality GUI designs suitable for both prototypes and final interfaces. Its vector-based environment ensures precise alignment, consistent spacing, and scalable visuals without loss of clarity. Components, color styles, and typography can be systematized for design consistency across screens. However, heavy or AI-generated visuals may slightly impact performance, and fine-tuning detailed textures requires manual effort. Overall, Figma's output is clean, modern, and ideal for both presentation and development handoff.

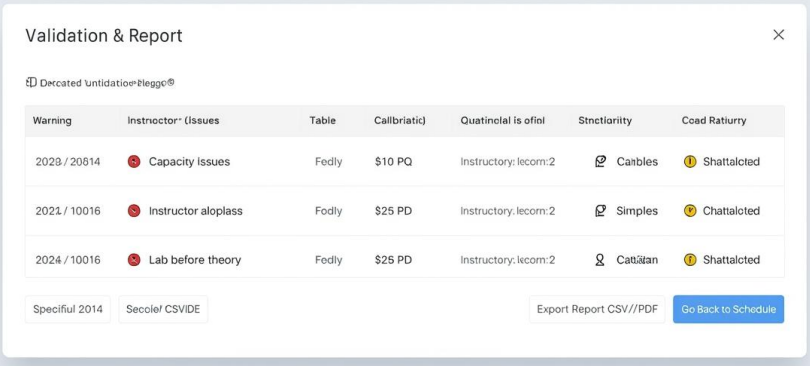
Usability: Figma's browser-based interface and collaborative features make it highly efficient for team-based GUI design. Real-time editing, comments, and shared libraries streamline communication and maintain design consistency. Its intuitive layout and plugin ecosystem allow easy prototyping, auto-layout creation, and design iteration. While advanced users benefit from extensive customization, beginners may initially find the tool complex. Overall, Figma combines powerful functionality with excellent usability for both design creation and collaboration.

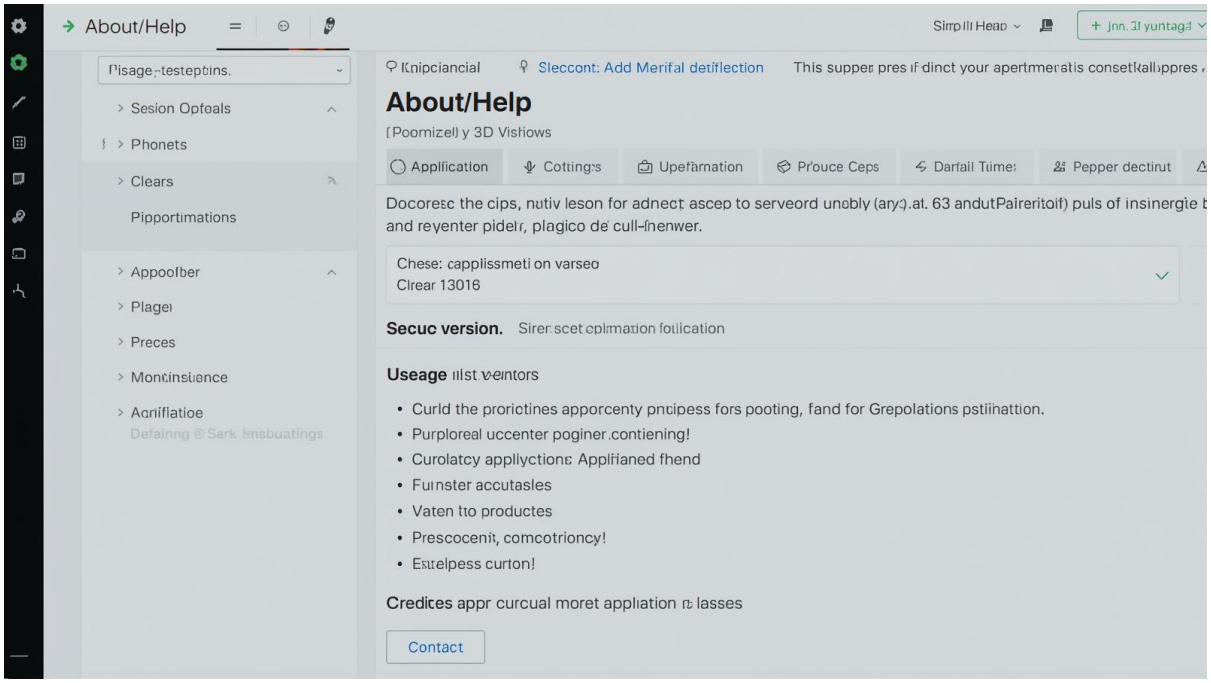
My versions of GUI:

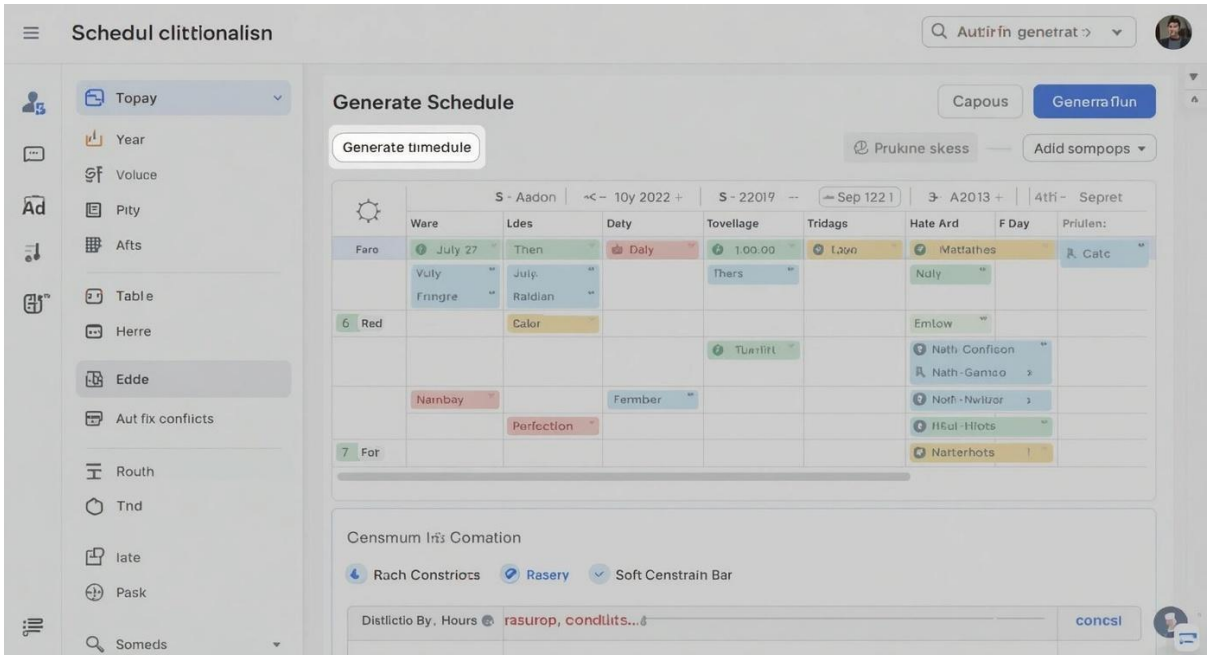
BEEplan
Forgot Password
ID
Secin
Login
Sign Up
Forgot Password


BeePlan – Course Scheduler
Manage Data
Generate Schedule
View Reports
View Reports
Clores In Crdet
Exit

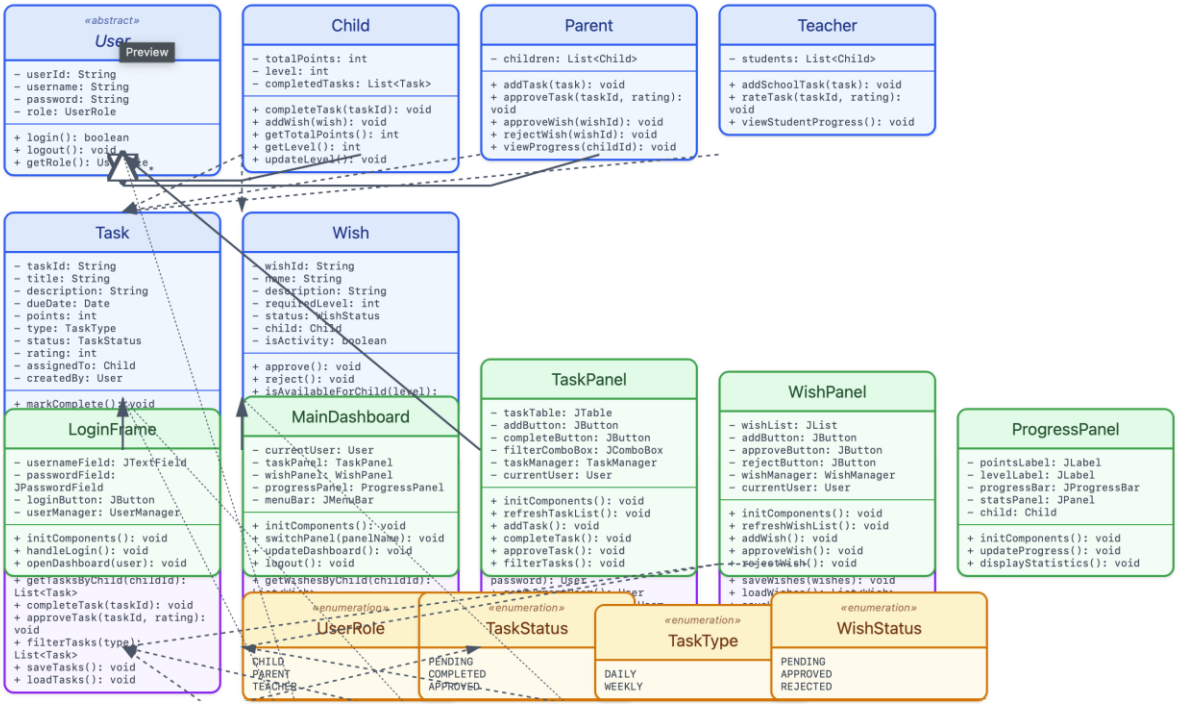

Data Management Page Concept
Coursees
Instructors
Rules & Labs
Clasmcators
Classrooms & Type
Add Course
Exities & Pnurales,
Add Courses
No Friday 13:20-15:10
Import JSov
Lab after Theory
Adhes Sureen


Validation & Report
Warning
Capacity issues
Instructor aloplass
Lab before theory
Table
Fedly
$10 PQ
$25 PD
$25 PD
Export Report CSV//PDF
Go Back to Schedule

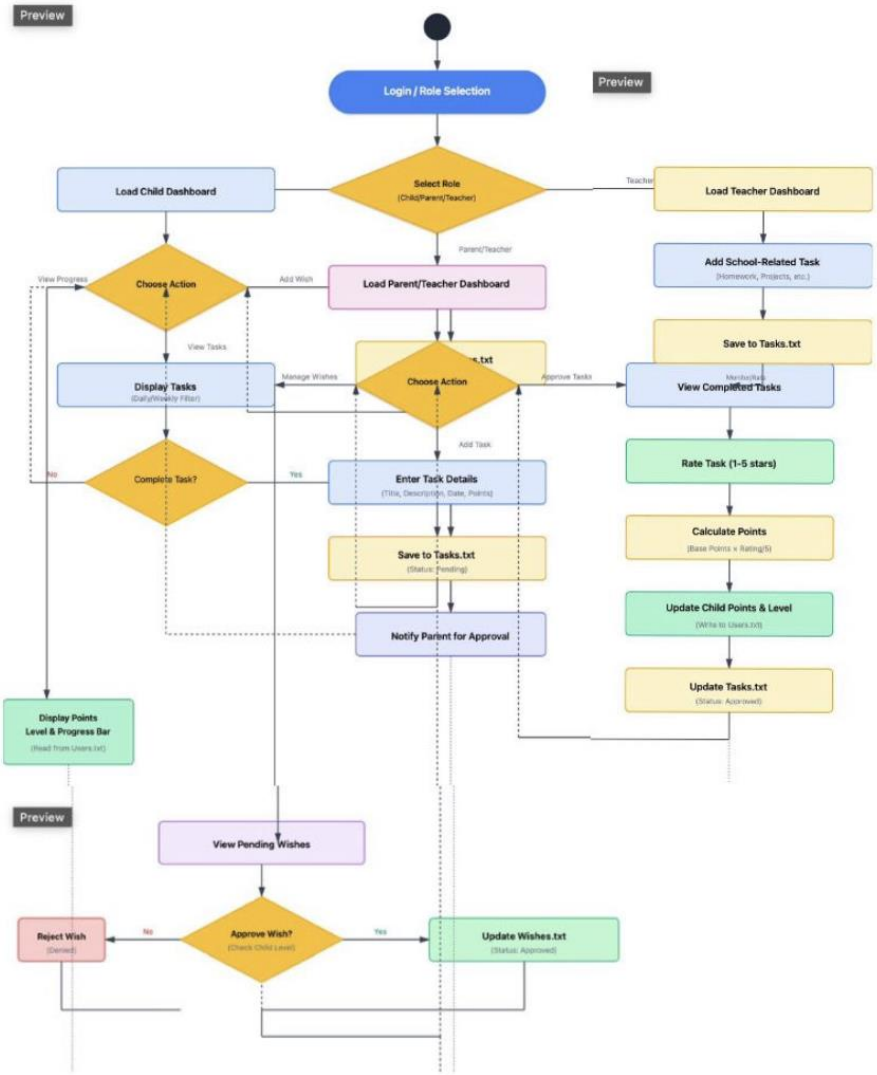


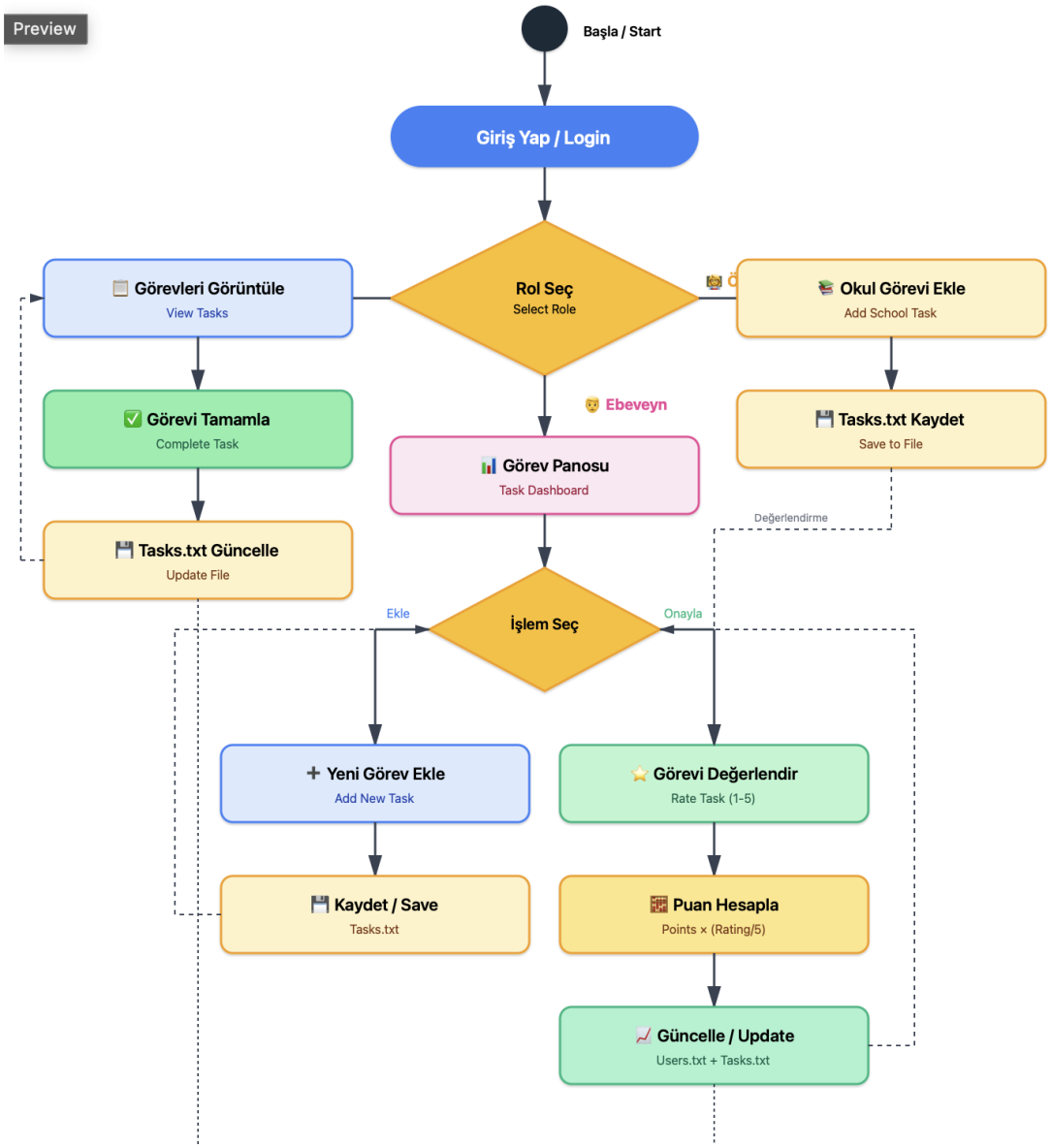


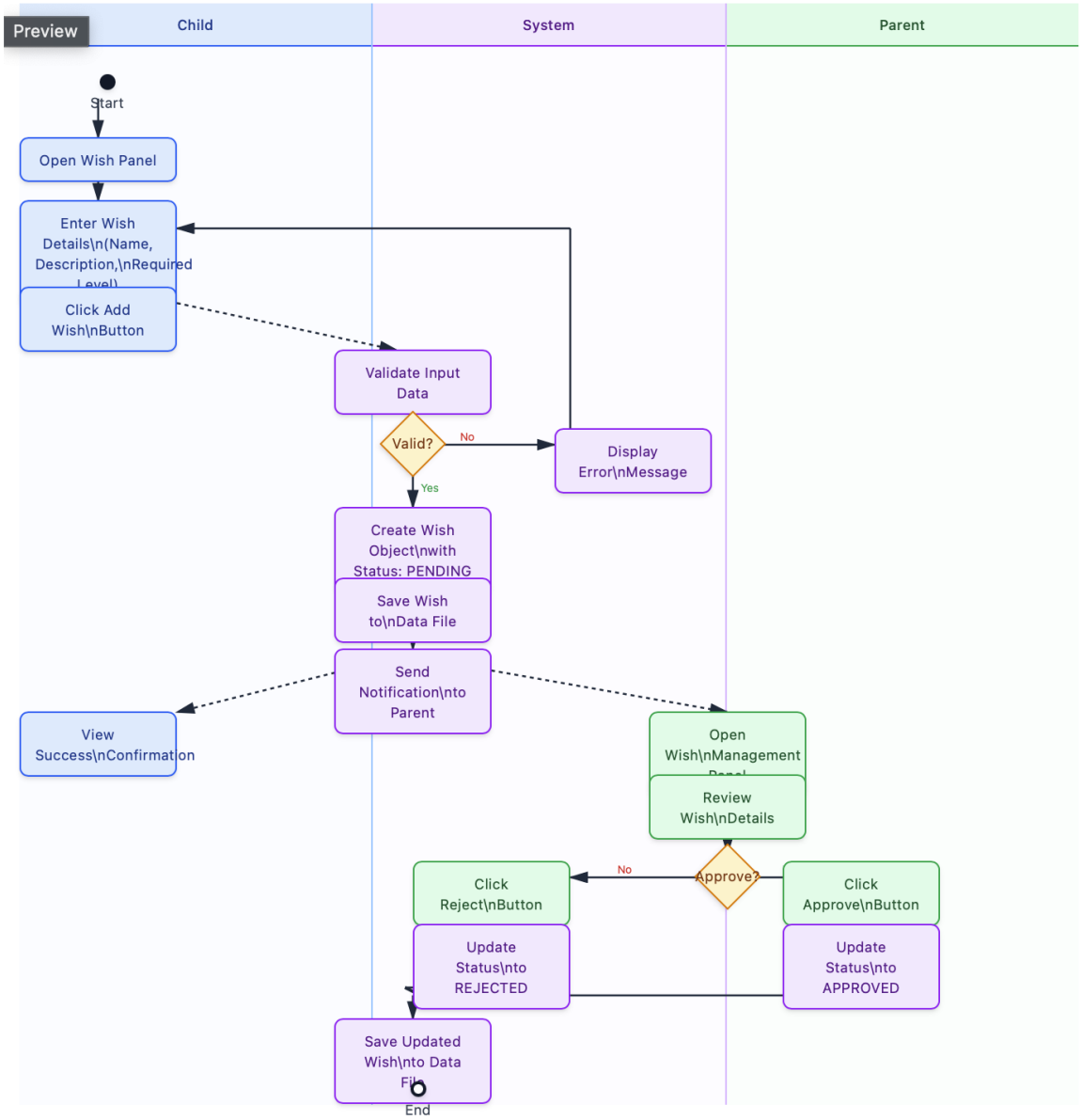
My versions of diagrams:


«abstract»
User
Preview
- userId: String
- username: String
- password: String
- role: UserRole
+ login(): boolean
+ logout(): void
+ getRole(): UserRole
Child
- totalPoints: int
- level: int
- completedTasks: List<Task>
+ completeTask(taskId): void
+ addWish(wish): void
+ getTotalPoints(): int
+ getLevel(): int
+ updateLevel(): void
Parent
- children: List<Child>
+ addTask(task): void
+ approveTask(taskId, rating): void
+ approveWish(wishId): void
+ rejectWish(wishId): void
+ viewProgress(childId): void
Teacher
- students: List<Child>
+ addSchoolTask(task): void
+ rateTask(taskId, rating): void
+ viewStudentProgress(): void
Task
- taskId: String
- title: String
- description: String
- dueDate: Date
- points: int
- type: TaskType
- status: TaskStatus
- rating: int
- assignedTo: Child
- createdBy: User
+ markComplete(): void
Wish
- wishId: String
- name: String
- description: String
- requiredLevel: int
- status: WishStatus
- child: Child
- isActivity: boolean
+ approve(): void
+ reject(): void
+ isAvailableForChild(level):
TaskPanel
- taskTable: JTable
- addButton: JButton
- completeButton: JButton
- filterComboBox: JComboBox
- taskManager: TaskManager
- currentUser: User
+ initComponents(): void
+ refreshTaskList(): void
+ addTask(): void
+ completeTask(): void
+ approveTask(): void
+ filterTasks(): void
WishPanel
- wishList: JList
- addButton: JButton
- approveButton: JButton
- rejectButton: JButton
- wishManager: WishManager
- currentUser: User
+ initComponents(): void
+ refreshWishList(): void
+ addWish(): void
+ approveWish(): void
+ saveWishes(wishes): void
ProgressPanel
- pointsLabel: JLabel
- levelLabel: JLabel
- progressBar: JProgressBar
- statsPanel: JPanel
- child: Child
+ initComponents(): void
+ updateProgress(): void
+ displayStatistics(): void
LoginFrame
- usernameField: JTextField
- passwordField: JPasswordField
- loginButton: JButton
- userManager: UserManager
+ initComponents(): void
+ handleLogin(): void
+ openDashboard(user): void
MainDashboard
- currentUser: User
- taskPanel: TaskPanel
- wishPanel: WishPanel
- progressPanel: ProgressPanel
- menuBar: JMenuBar
+ initComponents(): void
+ switchPanel(panelName): void
+ updateDashboard(): void
+ logout(): void
+ getWishesByChild(childId):
+ getTasksByChild(childId): List<Task>
+ completeTask(taskId): void
+ approveTask(taskId, rating): void
+ filterTasks(type): List<Task>
+ saveTasks(): void
+ loadTasks(): void
«enumeration»
UserRole
CHILD
PARENT
TEACHER
«enumeration»
TaskStatus
PENDING
COMPLETED
APPROVED
«enumeration»
TaskType
DAILY
WEEKLY
«enumeration»
WishStatus
PENDING
APPROVED
REJECTED

Complete System Activity Diagram




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
📊 Görev Panosu
Task Dashboard
💾 Tasks.txt Kaydet
Save to File
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


Preview
Child
System
Parent
Start
Open Wish Panel
Enter Wish Details\n(Name, Description,\nRequired
Click Add Wish\nButton
Validate Input Data
Valid?
No
Display Error\nMessage
Yes
Create Wish Object\nwith Status: PENDING
Save Wish to\nData File
Send Notification\nto Parent
View Success\nConfirmation
Open Wish\nManagement
Review Wish\nDetails
Approve?
No
Click Reject\nButton
Click Approve\nButton
Update Status\nto REJECTED
Update Status\nto APPROVED
Save Updated Wish\nto Data
End