

Quick Quacker: Sprint 1

System Design

Group 1:

- ☒ Phoebe Gamit
- ☒ Kristin Villena
- ☒ Michael David
- ☒ Amnol Misra
- ☒ Urgyen Kalsang

Table of Contents

Table of Contents	2
CRC Cards	3
Duck	
Game Manager	
UI Manager	
Storage Manager	4
Task	
Task Manager	5
XP Manager	
Achievement Manager	
Software Architecture Diagram	6

CRC Cards

Duck	
<p>Maintain its name, level, XP, hunger, thirst, and mood.</p> <p>Respond to user actions like feeding, giving water, and playing.</p> <p>Display animations and visual changes based on interactions.</p> <p>Level up when XP reaches a threshold.</p>	<p>GameManager (Handles interactions)</p> <p>UIManager (Updates the display)</p> <p>StorageManager (Saves and loads duck data)</p>

Game Manager	
<p>Control game logic (e.g., hunger and thirst decrease over time).</p> <p>Process user inputs (e.g., feeding, playing).</p> <p>Handle XP gain and level-up logic.</p> <p>Unlock achievements.</p>	<p>Duck (Updates the pet's stats)</p> <p>UIManager (Updates the UI)</p> <p>AchievementManager (Tracks milestones)</p>

UI Manager	
<p>Render duck stats and animations.</p> <p>Update XP, hunger, thirst, and level displays.</p> <p>Manage settings and modal interactions.</p> <p>Display achievements when unlocked.</p> <p>Render task list.</p> <p>Display filter options (All, Active, Completed).</p>	<p>GameManager (Receives updates)</p> <p>Duck (Displays status)</p> <p>TaskManager (Fetches and updates tasks)</p> <p>Task (Displays individual tasks)</p> <p>XPManager (Updates progress bar based on task completion)</p>

Provide an interface for adding/editing tasks. Update task stats and XP level.	
---	--

Storage Manager	
Save and load duck stats using LocalStorage. Retrieve stored progress when the game starts. Store achievements and preferences. Save and retrieve tasks from LocalStorage. Persist user preferences and XP progress. Load saved tasks when the app starts.	GameManager (Handles save/load logic) UIManager (Loads settings on startup) TaskManager (Manages tasks) XPManager (Stores XP progress)

Storage Manager	
Save and load duck stats using LocalStorage. Retrieve stored progress when the game starts. Store achievements and preferences.	GameManager (Handles save/load logic) UIManager (Loads settings on startup)

Task	
Store task details (title, priority, completion status). Toggle completion status.	TaskManager (Manages task operations) UIManager (Displays tasks)

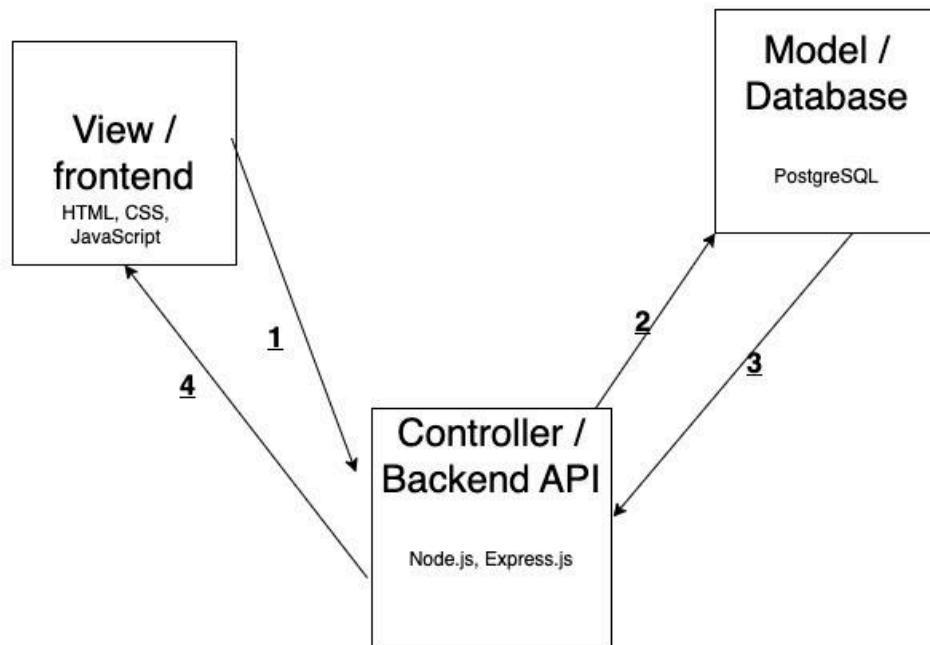
Provide information for rendering in UI.	
--	--

Task Manager	
Add, remove, and edit tasks. Toggle task completion. Filter tasks based on status (Active, Completed). Sort tasks by priority. Save and load tasks from LocalStorage.	TaskManager (Manages task operations) UIManager (Displays tasks) Task (Handles individual task data) UIManager (Updates UI based on task operations) StorageManager (Persists tasks)

XPManager	
Track XP level based on completed tasks. Update and display XP progress bar. Unlock achievements.	TaskManager (Earns XP from task completion) UIManager (Displays XP progress) AchievementManager (Unlocks rewards)

AchievementManager	
Track and unlock task-based achievements. Display notifications for new achievements. Store unlocked achievements.	XPManager (Triggers achievements based on XP milestones) UIManager (Displays achievements)

Software Architecture Design



1. The **View (Frontend)** sends requests to the **Controller (Backend)**.
2. The **Controller** processes the request and interacts with the **Model (Database)**.
3. The **Model** stores/retrieves data and sends responses back to the **Controller**.
4. The **Controller** sends the final response to the **View**.