# DesignGenie – Sprint 3 Final Report

**Course:** Fundamentals of Software Engineering
**Project Name:** DesignGenie
**Group Members:**


- Urwa Khalid (23L-2573)
- Faryal Zahra (23L-2551)
- Areeba Noor (23L-2562)

## a) Project Introduction

DesignGenie is an AI-powered web application that streamlines the design creation process for professionals and students in fields such as architecture, product design, and prototyping. The platform allows users to input specific constraints (like size, shape, and material), and then generates optimized design suggestions using AI.

The user interface is structured as follows:

- A **"Get Started"** screen directs users to a **design category selection** (Architecture, Product, or Prototype).
- After selecting a category, users enter input constraints.
- The system validates input and, if correct, generates a visual design using backend logic.
- Users are presented with **options to regenerate or download** the final design.

DesignGenie minimizes repetitive manual design work and enables faster ideation, optimization, and exporting of design outputs.

## b) User Stories (All Sprints)

**Sprint 1 – Interface & Navigation**

- Setup of initial UI with navigation
- Form for entering design constraints
- Static previews for early testing

**Sprint 2 – Core Functionality**

- **US-03: Generate AI-Based Design**
  *As a designer, I want to automatically generate design options based on constraints so I can explore creative possibilities.*
- **US-04: Optimize AI-Based Design**
  *As a designer, I want the system to improve my selected design to enhance quality and efficiency.*

**Sprint 3 – Output Features**

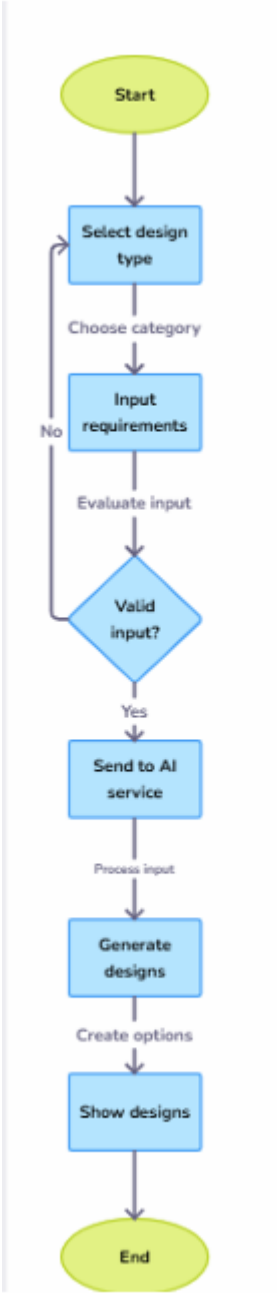
- **US-05: Export Final Design**
  *As a user, I want to export my final design in PNG format for saving or sharing.*

- **US-06: Regenerate Designs**
  *As a user, I want to regenerate designs using updated inputs so I can see different options without restarting.*

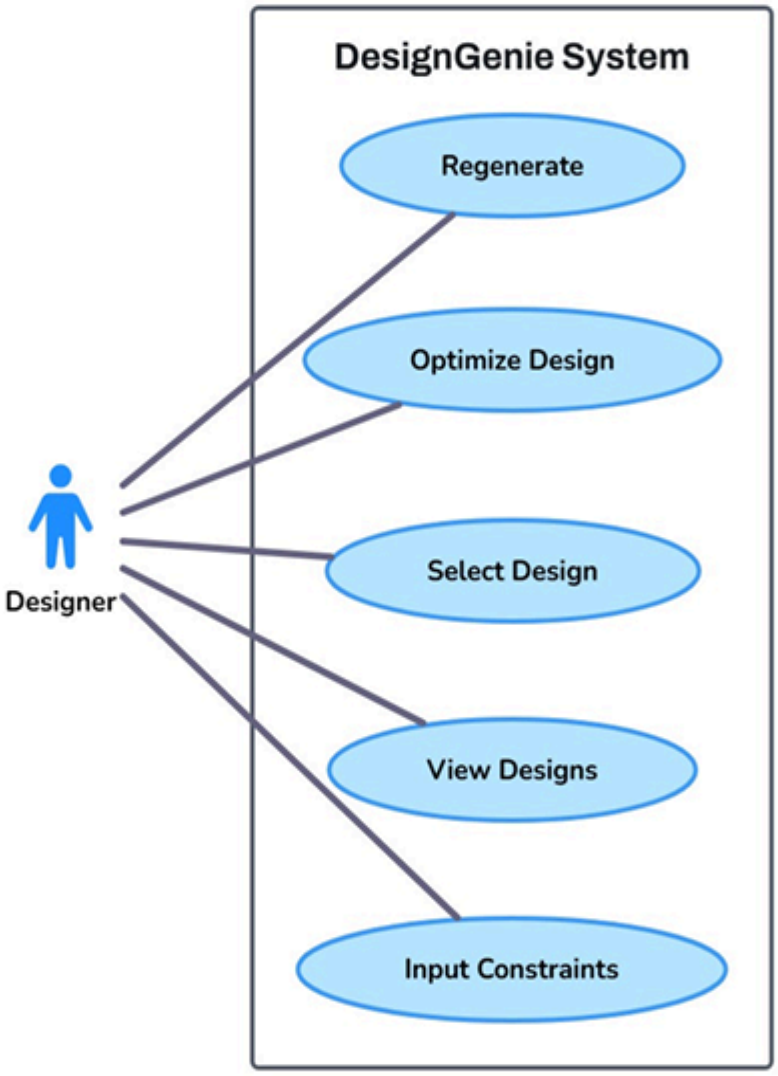
## c) Design (Sprint 2 Items)

During Sprint 2, the team created detailed design artifacts to represent system logic and user flow:

- **Activity Diagram:**
    - Constraint Input → Generate → Optimize → Export
    - User flow from selecting category to receiving design output

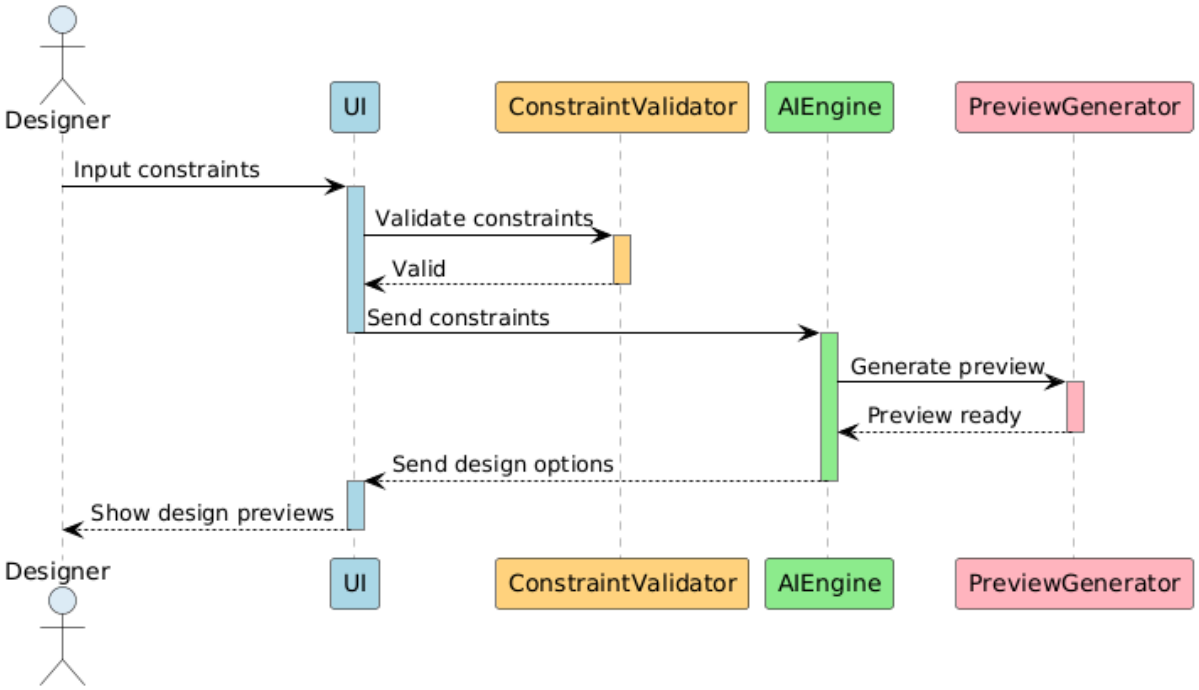



- **Use Case Diagram:**
  - Actor: Designer
  - Use Cases: Input Constraints, Generate Design, Optimize, Regenerate, Export

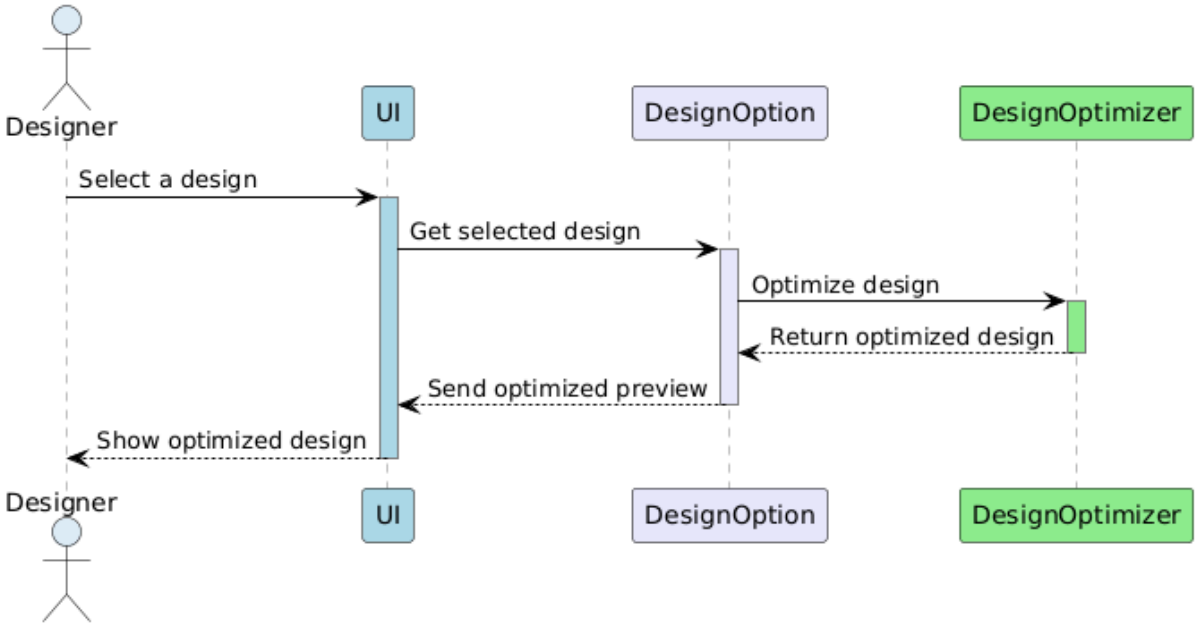



- **Sequence Diagram:**

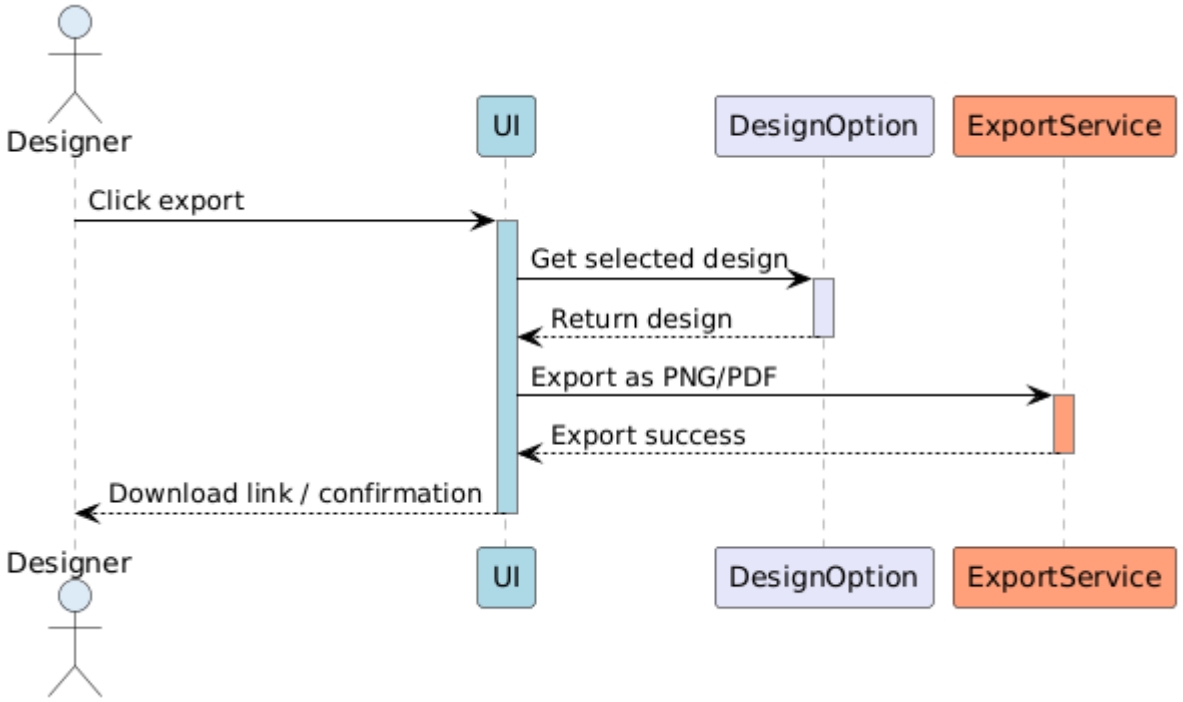
1.Input Constraints → Trigger AI Generation



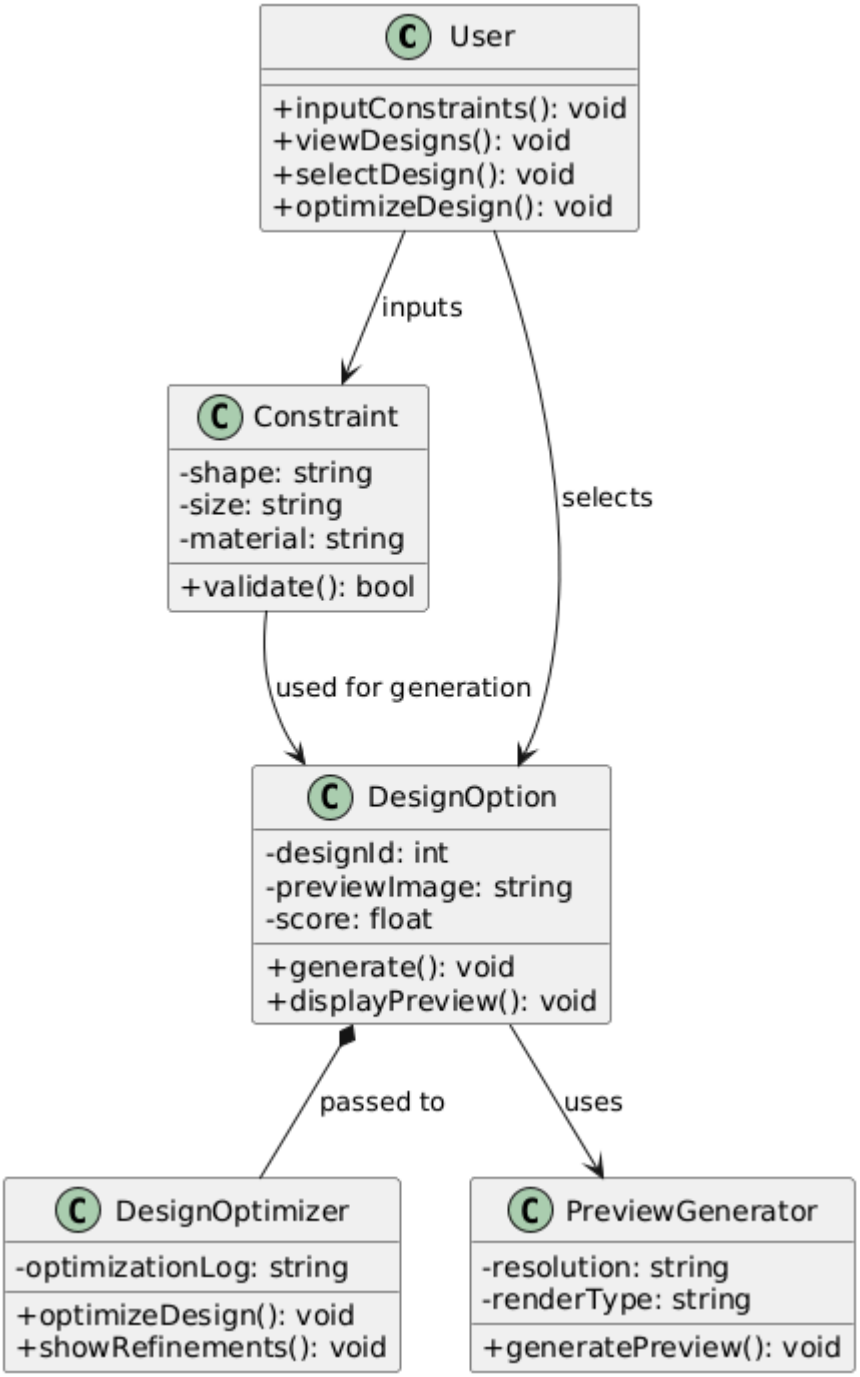

2.Select Design → Optimize




3.Select Design → Optimize




- **Class Diagram:**
    - Key Classes: `User`, `Constraint`, `DesignOption`, `DesignOptimizer`, `ExportHandler`




These designs helped guide the development of a consistent and modular application.

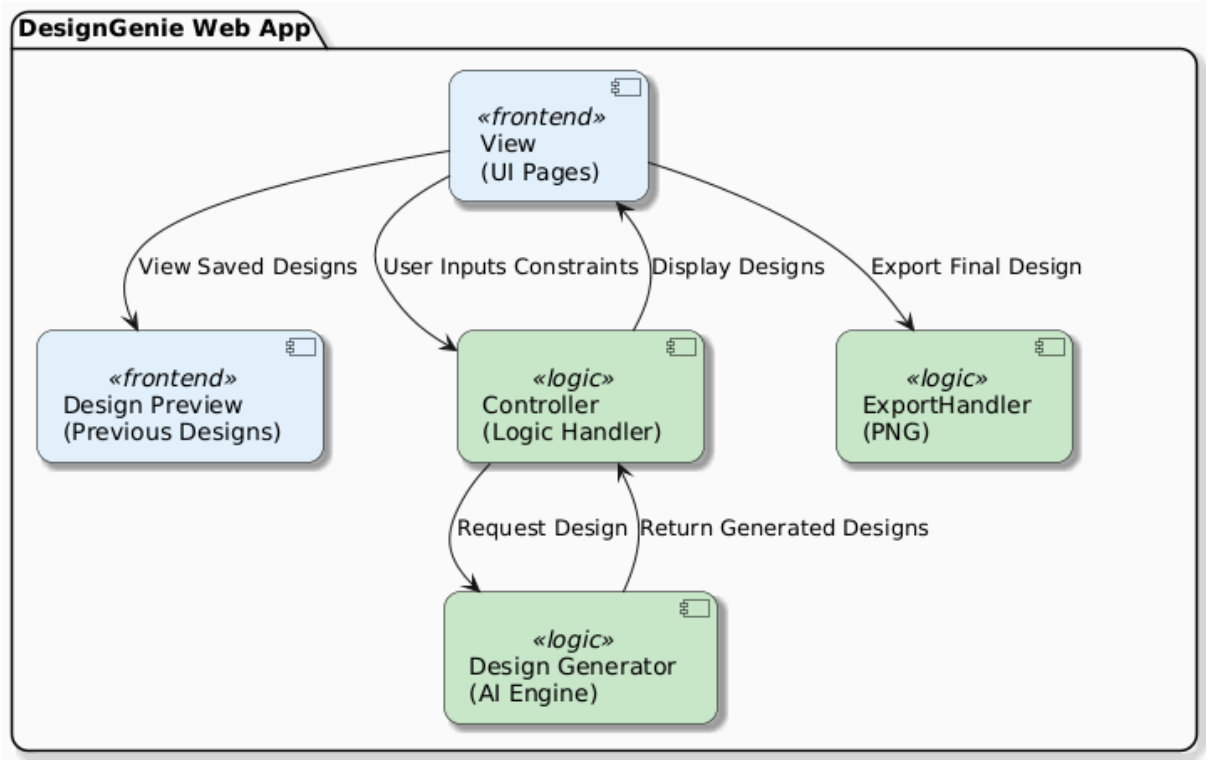
## d) Architecture

We implemented the **Model-View-Controller (MVC)** architecture to ensure modularity, scalability, and separation of concerns.

**Components:**

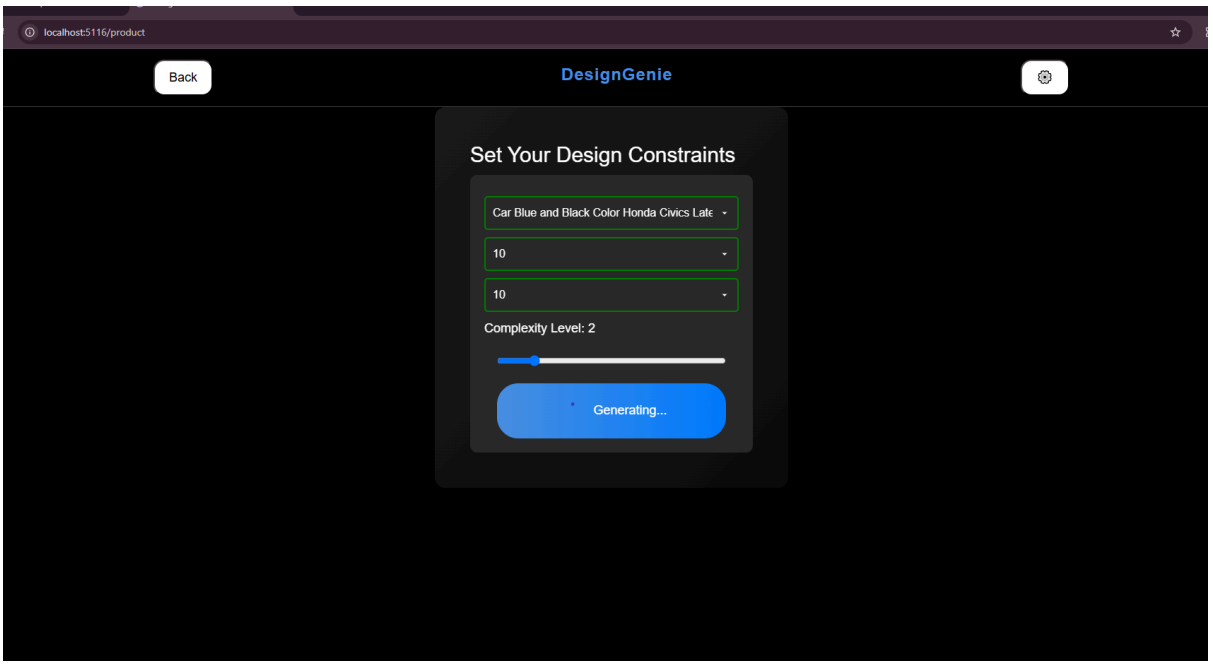
- **Model**: Stores constraint data and executes AI-based design generation logic.
- **View**: The graphical user interface (GUI) where users interact with the app.
- **Controller**: Bridges the input/output operations with the underlying business logic.

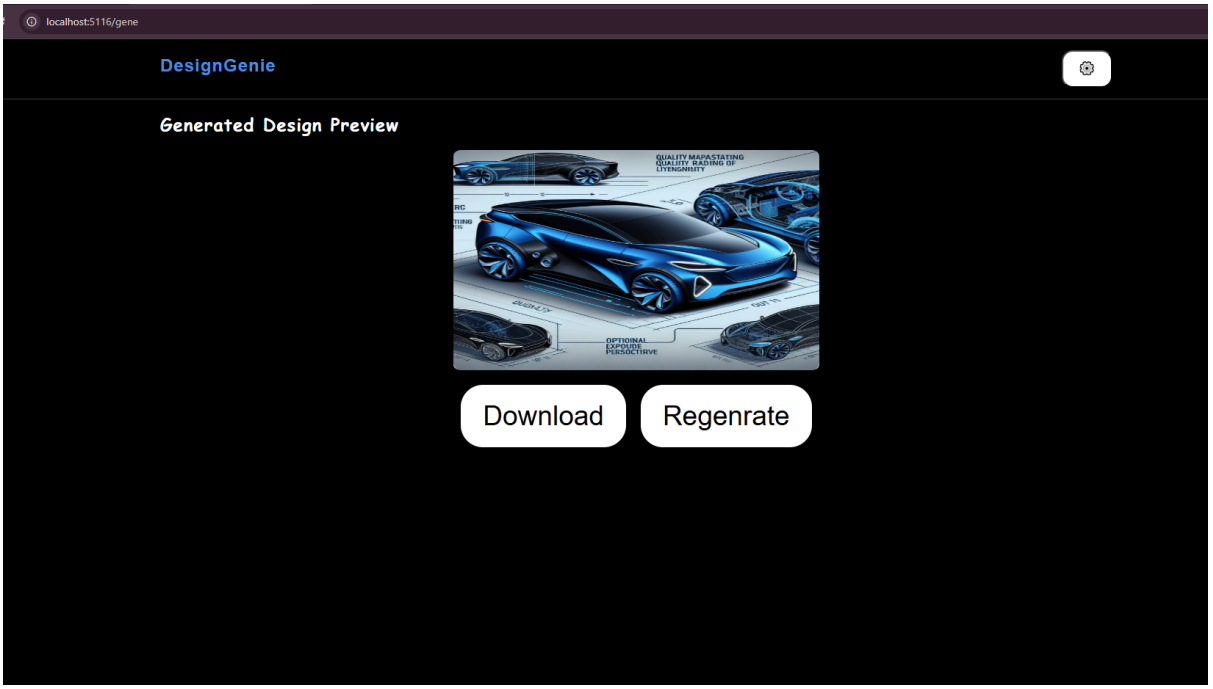
**Architecture Diagram:**




This pattern allowed parallel development of frontend and backend modules with clean integration.

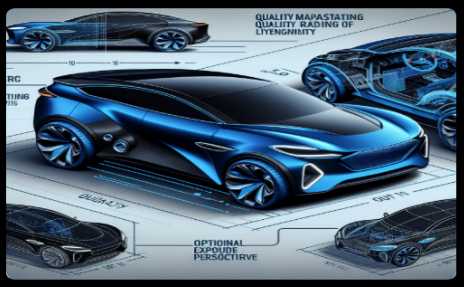
## e) Actual Implementation Screenshots:


localhost:5116/product
Back
DesignGenie
Set Your Design Constraints
Car Blue and Black Color Honda Civics Late
10
10
Complexity Level: 2
Generating...

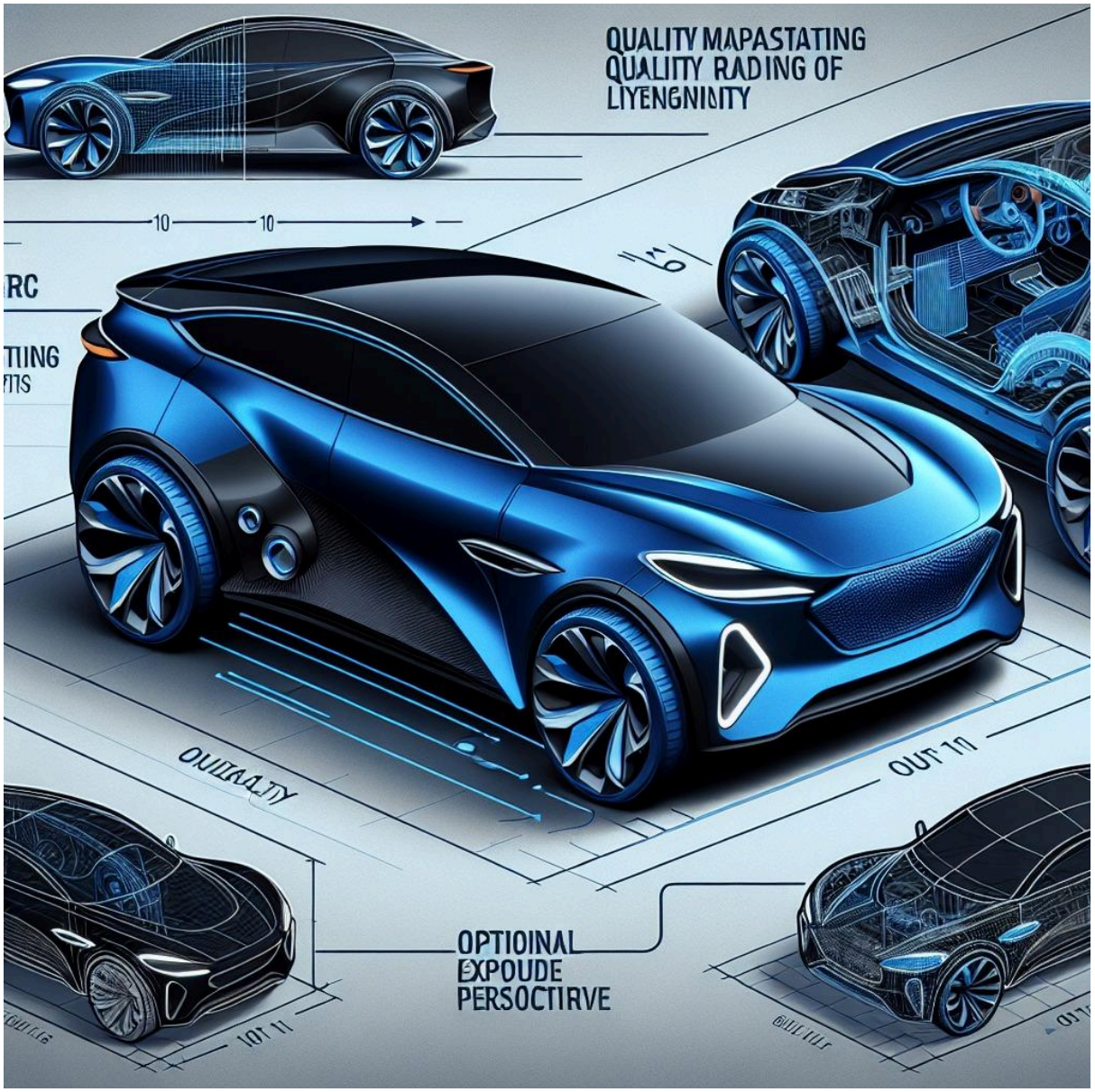

localhost:5116/gene
DesignGenie
Generated Design Preview
Download
Regenrate

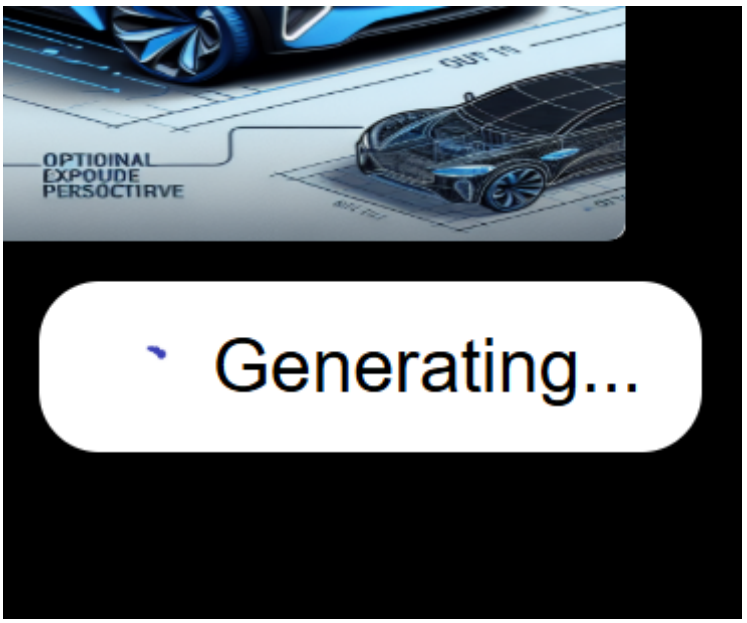
generated-image (2).png
1,944 KB • Done

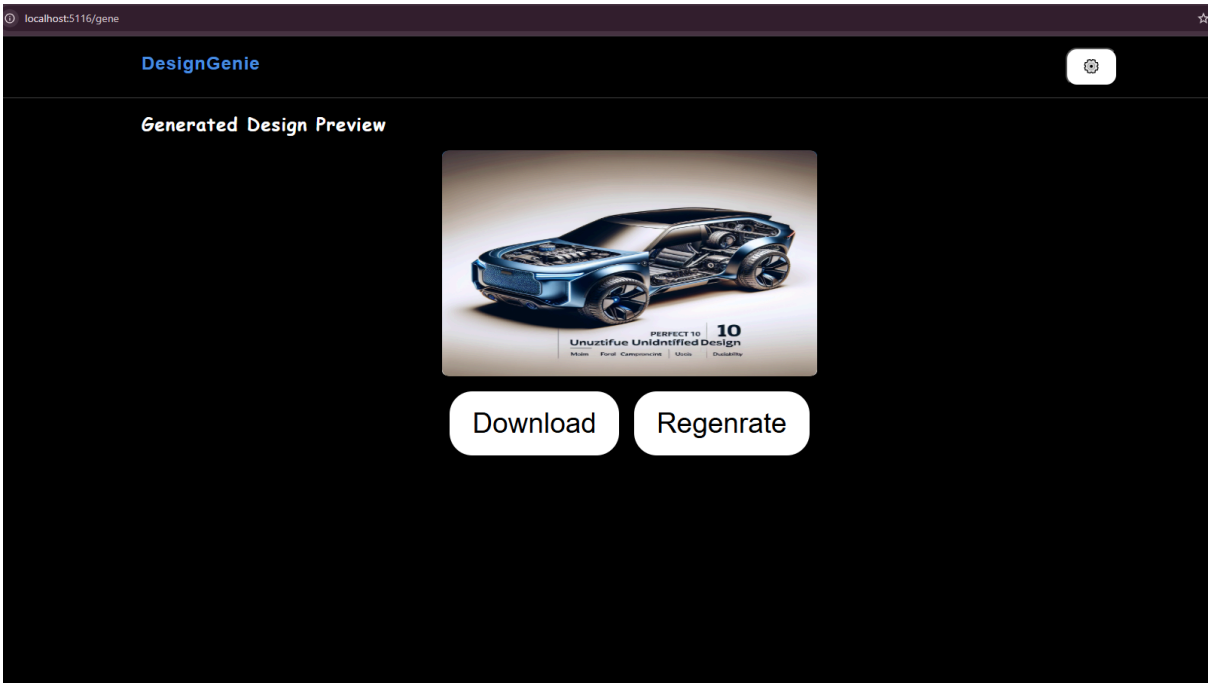
view




Download Regenrate

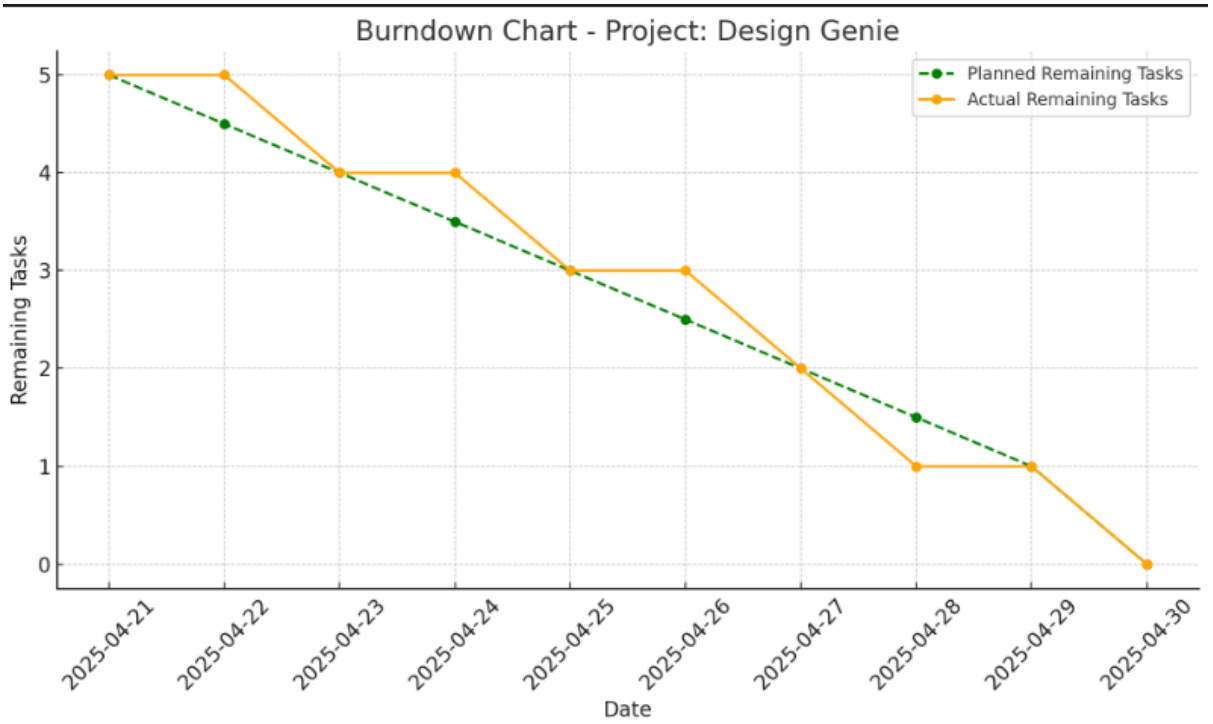

QUALITY MAPASTATING
QUALITY RADING OF
LIYENGNIITY
10
10
RC
TTING
OPTIOINAL
EXPOUDE
PERSOCTIRVE



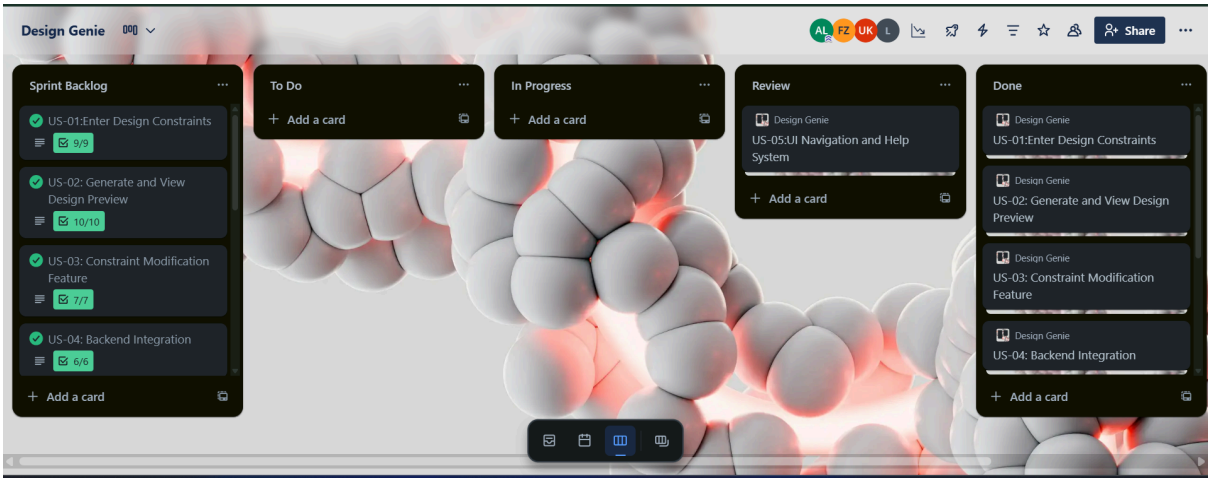




## f) Product Burndown Chart

Below is the burn down chart showing how tasks were completed across the Sprint 3 timeline:




**g) Trello Board Screenshot:**




---

**h) Boundary Value Analysis Testing**Since our application does not include login or signup, we applied **Boundary Value Analysis (BVA)** to input fields used for constraint-based design generation.

Boundary Value Analysis (BVA) Test Cases

1. Architecture Design Constraints

| Test Case | Architecture Name | Width | Height | Material | Expected Result |
|---|---|---|---|---|---|
| 1 | “” | 20 | 30 | Wood | Invalid |
| 2 | “Chair” | -1 | 30 | Wood | Invalid |
| 3 | “Chair” | 0 | 30 | Wood | Invalid |
| 4 | “Chair” | 10 | 30 | Wood | Valid |
| 5 | “Chair” | 30 | 0 | Wood | Invalid |
| 6 | “Chair” | 30 | 10 | Wood | Valid |
| 7 | “Chair” | 30 | 30 | --Select-- | Invalid |

2. Product Design Constraints

| Test Case | Product Name | Quality | Size | Expected Result |
|---|---|---|---|---|
| 1 | “” | 5 | 20 | Invalid |
| 2 | “” | 0 | 20 | Invalid |
| 3 | “Car” | 10 | 20 | Valid |
| 4 | “Car” | 10 | 35 | Valid |
| 5 | “Car” | 11 | 35 | Invalid |
| 6 | “Car” | 5 | 0 | Invalid |
| 7 | “Car” | 2 | 10 | Valid |

3. Prototype Info Constraints

| Test Case | Project Name | Width | Height | Expected Result |
|---|---|---|---|---|
| 1 | "" | 200 | 300 | Invalid |
| 2 | “Task Management System” | 200 | 300 | Invalid |

| | | | | |
|---|---|---|---|---|
| 3 | “Task Management System” | 99 | 300 | Invalid |
| 4 | “Task Management System” | 1000 | 1000 | Valid |
| 5 | “Task Management System” | 5000 | 3000 | Valid |
| 6 | “Task Management System” | 5001 | 300 | Invalid |
| 7 | “Task Management System” | 300 | 99 | Invalid |
| 8 | “Task Management System” | 3000 | 1000 | Valid |
| 9 | “Task Management System” | 3000 | 5000 | Valid |
| 10 | “Task Management System” | 300 | 5001 | Invalid |

These tests helped validate all important input paths and improved form reliability.

## i) Work Division Between Group Members:

| **Member** | **Role** | **Responsibilities** |
|---|---|---|
| **Urwa Khalid** | Backend Developer & Team Lead | AI generation, regeneration logic, export functionality |

| | | |
|---|---|---|
| **Faryal Zahra** | UI/UX Developer | Design category UI, validation messages, regeneration UI |
| **Areeba Noor** | QA & Documentation Lead | Testing (BVA), Trello task tracking, report preparation |

We followed an agile workflow with clearly assigned roles and collaborative feedback.

## j) Lesson Learned by Group

- **Importance of BVA**: Testing real inputs rather than assumed login systems taught us how to tailor QA to actual system needs.
- **Modularity helps scaling**: MVC makes implementation easier and cleaner.
- **Regeneration adds value**: A small feature that gives the user much more flexibility.
- **Team tools matter**: Trello and GitHub were crucial in tracking progress and syncing code.
- **UI/UX impacts everything**: Input validation and visual feedback directly affected user satisfaction.

## Benefits of DesignGenie

DesignGenie brings several key benefits to its users and developers, making it a practical and impactful AI-powered solution:

**1. Accelerates the Design Process**
DesignGenie eliminates the need for manual sketching or repeated trial-and-error by generating optimized designs based on user-defined constraints. This significantly reduces the time spent on ideation and early prototyping, especially for students and professionals who work under tight deadlines.

**2. Enhances Creativity and Exploration**
By providing multiple design options with the ability to regenerate suggestions, users are encouraged to explore creative possibilities they might not have initially considered. The AI's capability to produce diverse outputs enhances innovation and broadens design thinking.

**3. Improves User Experience through Smart Validation**
Input validation and visual feedback ensure users are guided throughout the process, reducing errors and making the experience more intuitive and user-friendly.

**4. Promotes Reusability and Scalability**
Thanks to the modular MVC architecture, the system can be easily expanded in the future. For

example, new design categories or AI models can be integrated without overhauling the core structure, making it scalable for broader use cases.

**5. Real-World Application of Software Engineering Principles**

For the development team, DesignGenie was a practical application of software engineering concepts such as Agile development, boundary value analysis (BVA), modular architecture, and effective team collaboration. These experiences are directly transferable to real-world software development environments.

**6. Supports Remote Collaboration and Project Management**

By using tools like Trello and GitHub, the team demonstrated efficient task tracking, collaboration, and version control. This not only improved project delivery but also prepared the team for industry-level software workflows.

**7. Accessible Output Sharing**

With export functionality (e.g., PNG format), users can easily save and share their designs with peers, clients, or instructors, enabling smoother communication and review processes.

## Conclusion

DesignGenie has evolved over the course of three sprints into a fully functional, user-friendly web application that simplifies the design generation process using AI. By integrating a structured approach through the MVC architecture, focusing on practical testing with boundary value analysis, and maintaining strong task coordination via Trello, we successfully delivered all planned features.

From the initial "Get Started" interaction to constraint-based input validation and image generation with options for regeneration and download, the application reflects thoughtful design and solid implementation. This project not only helped us apply software engineering principles effectively but also gave us hands-on experience with real-world development cycles. The collaborative journey strengthened our technical and project management skills, setting a strong foundation for future work in agile environments.