ST{AU}CKTION

Ahmet Oğuz Ergin - 21903070
Ayşe Özge Erkan - 21902107
Bertan Özer - 21902475
Damla Köklü - 22003189
Göktuğ Yeşilyurt - 22003697

Team 10

Project Identification

Project Name: St{au}cktion
Product Brand Name: St{au}cktion
Supervisor: Cüneyt Sevgi

Team Members and Roles:
Ayşe Özge Erkan: Front-End Developer
Bertan Özer: Back-End Developer
Ahmet Ergin: Back-End Developer
Damla Köklü: Front-End Developer
Göktuğ Yeşilyurt: Back-End Developer

Project Management: Responsibilities rotate monthly among all team members.
Collaboration: Everyone participated in testing to ensure cross-functional quality assurance.

Software Product Identification

-The purpose is to provide photographers a platform they can use to sell their photos to stock image companies or other end users looking for stock images, allowing anyone to purchase the photos conveniently, and give photographers more control over the price.

-Compared to alternatives such as Getty Images, our project provides a platform that enables photographers of all skill levels to showcase their work.

-Our platform gives photographers more control over the price

The "What" of the Product

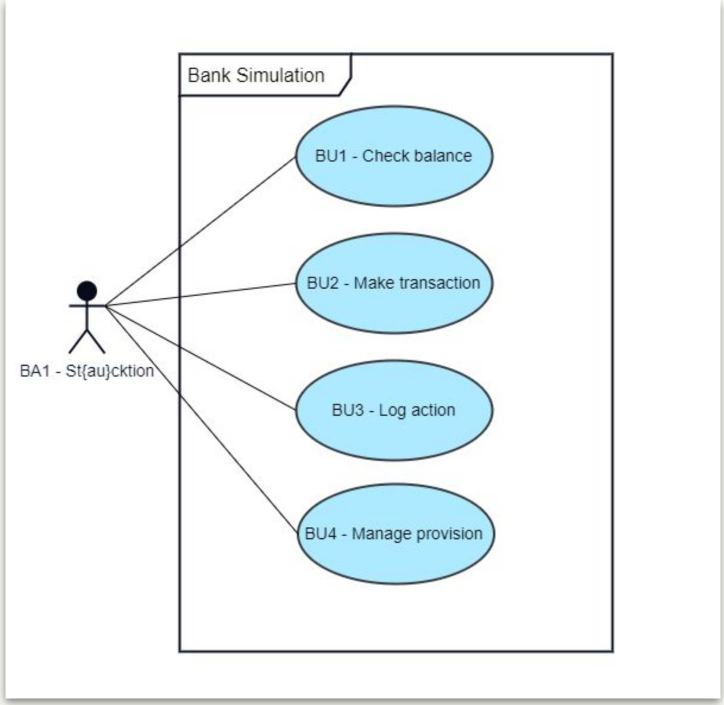
-Enable photographers to upload their photos directly onto the platform.
-Allow photographers to set and adjust their pricing.
-Facilitate auctions where stock image companies and other potential buyers can bid on photos.
-Provide a voting system for the community to highlight popular photos, enhancing visibility for high-quality work.

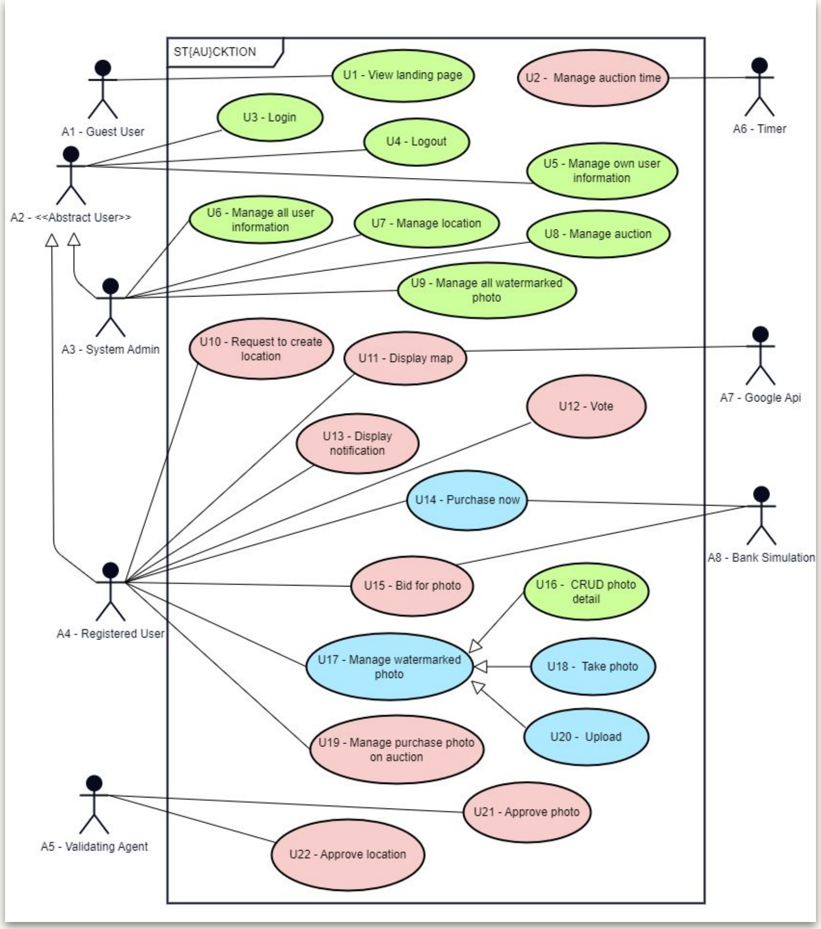
The "How" of the Product

-Photographers register on the platform and upload their photos with the option to set initial asking prices.
-Photos undergo a validation process to ensure they meet quality and content standards.
-Once approved, the photos become available for viewing and bidding in an auction format.
-Stock image companies and other buyers can browse, vote, and bid on these photos during auction periods.
-Users can vote for their favorite photos, influencing which photos appear more prominently in searches or featured listings.

Allocation of the Use Cases to 3 Increments

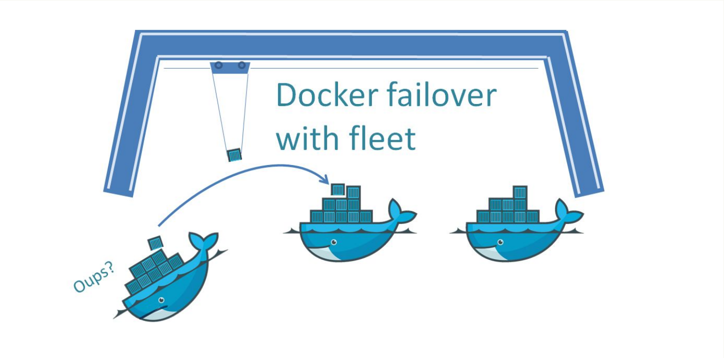
🔵 - 1 🟢 - 2 🔴 - 3



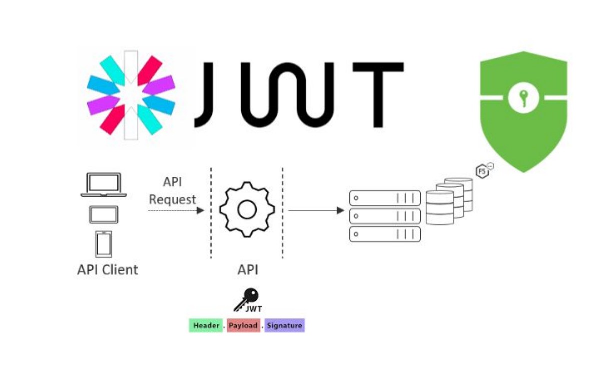


Non Functional Requirements

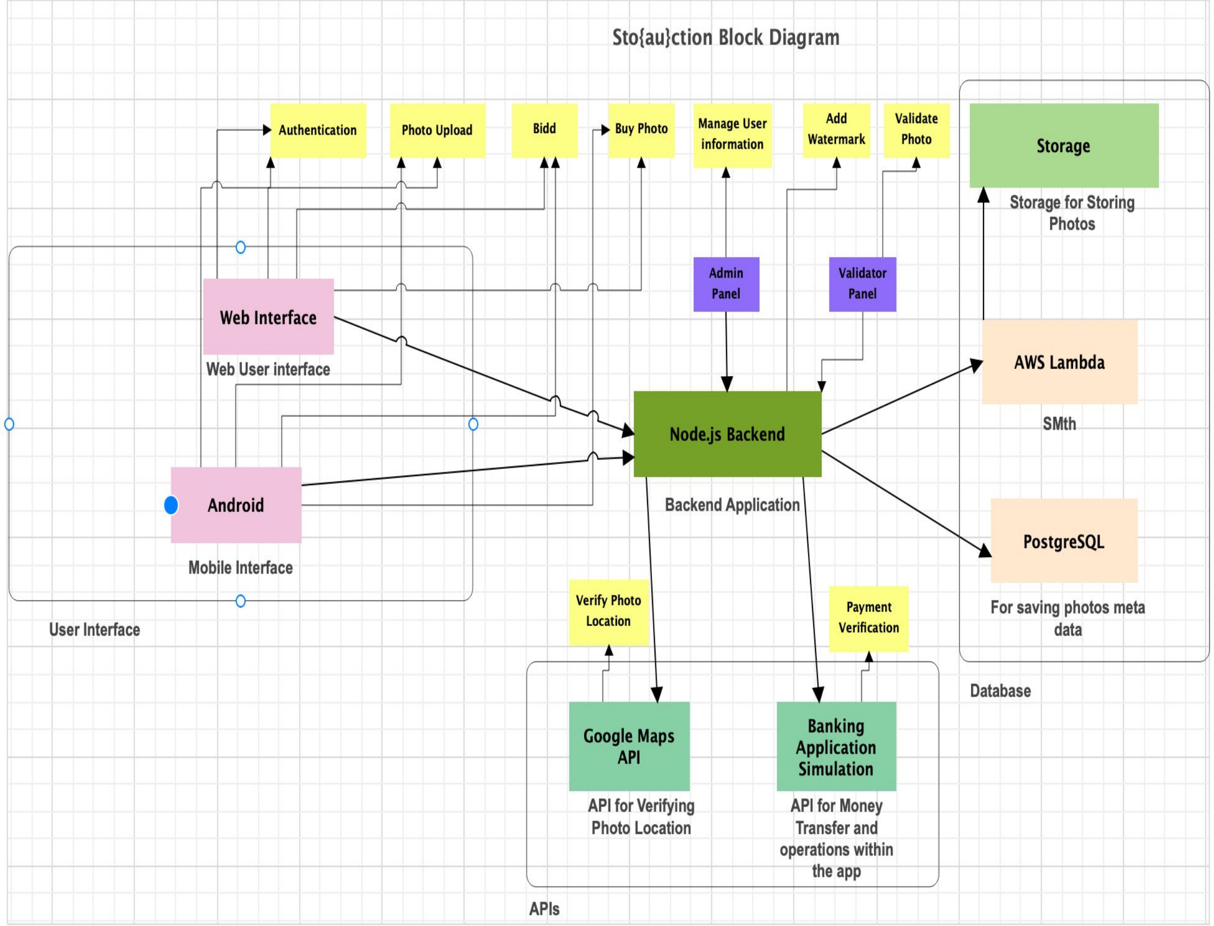
NReq7.0: The system architecture shall replace failed system modules with healthy ones automatically.



NReq8.0: Data transactions using tokens should be secured with the HS256 algorithm for integrity.



Block Diagram



Category	Technology/Tool	Purpose
Programming Languages	TypeScript, Java, Kotlin	Web development (TypeScript) and Android application development (Java and Kotlin)
Frameworks	React	Web User Interface development
Libraries	Sharp, Multer	Image storage (Multer) and image processing (Sharp)
Databases	PostgreSQL	Main data storage solution
APIs	Google Maps API, Simulated Banking API	Location information (Google Maps API) and bank transaction process (Simulated Banking API)
Cloud Services	AWS Lambda	Serverless computing for heavy tasks like image watermarking
Web Servers	Nginx	Routing and handling requests
Verification & Testing	Postman, SonarQube	API testing (Postman) and static code analysis (SonarQube)
Version Control	GitHub	Collaborative version control and repository management

Thank you For Listening

Team 10