Engineering of AI Systems

Car Detector Documentation

Group Members:

Name	Student ID
Nico Filzmoser	K12006215
Jose Juan Hernandez	K12330957

Table of Contents

1. Goal of the system

Q1. What is the domain of your system?

The domain of the system is image classification in the automotive consumer area.

Q2. Who would be the users of your system?

The users of the system would be anyone interested in buying, renting, or acquiring a car whose primary concern is to find alternative cars based on how the car looks.

Q3. What is the main goal of your system?

The main goal of the system is to provide an easy-to-use platform where users can upload photos of a car, have the car model identified, and then view a list of similar cars. Users can then further filter this list based on preferences for price range, etc...

Q4. How would your system achieve its goal?

The system would achieve its goal by utilizing a machine learning model trained on a dataset of car images to classify the car model from the uploaded photos. The system then generates a "fingerprint" by extracting the latent space of the AI model and compares it to the "fingerprint database" to determine a list of the most similar cars.

Q5. Which type of AI/ML strategy would be required/useful and why?

The AI/ML strategies required for this system would include:

- **Computer Vision**: To process and analyse the uploaded car images and classify the car model accurately. This would involve training a convolutional neural network (CNN) or similar on a diverse dataset of car images.
- **Image Classification**: To identify the specific make and model of the car from the uploaded images. This would be the core functionality enabling the system to provide relevant spare parts information.
- **Database Management**: While not an AI/ML strategy, efficiently managing and querying a database of similar cars is crucial for restricting the search based on user preferences.

These strategies are essential for ensuring that the system can accurately identify car models and provide users with the correct recommendations of similar cars.

2. Requirements

2.1 Goals of the system

Non-AI goals

User Interface

- Provide an intuitive and user-friendly platform where users can easily upload photos of cars.
- Presenting a list of cars that are visually similar to the input car.

Database Integration

- Maintain a comprehensive database of car models and their respective images for comparison.

Filtering Options

- Allow users to filter search results based on preferences such as price range, model year and / or brand.

AI goals

Image Classification: Accurately identify the make and possibly the model of the car from the uploaded images.

- Analysing images to identify car models and assess visual similarity is a complex task that is difficult to achieve through traditional programming methods alone. AI, specifically computer vision techniques, is designed for such tasks.

Similarity Matching: Generate a "fingerprint" of the car image and compare it with a database to find similar cars.

- Manually comparing car images for similarity would be incredibly time-consuming and not scalable for a platform intended for general use. AI enables automated and efficient processing of a large number of images and car data.
- Defining and quantifying visual similarity programmatically without AI would be challenging and subjective. Machine learning models can learn complex patterns of visual similarity from data, providing a more objective and consistent assessment.

2.2 Stakeholders

Users

- **Car Buyers/Renters**: Individuals looking to purchase or rent a car and are interested in exploring visually similar alternatives to cars they find appealing.
- **Car Enthusiasts**: People who are generally interested in cars and might use the system to discover new models based on visual preferences or for comparative purposes.

People Affected by the System

- **Car dealerships:** Car dealerships and manufacturers who might use the system to recommend alternative models to customers.
- **Car Marketplaces**: Online car marketplaces that integrate this system for enhanced user experience.
- **Automotive Review Blogs**: Could potentially use the system to enhance their content by providing visual similarity comparisons of cars.

Managers

- **Course Instructors**: For the purpose of this assignment, the instructors are the managers who will evaluate the system's design and requirements.

Regulators

- **Data Privacy Regulators**: Data protection agencies ensuring compliance with privacy laws.
- **Automotive Industry Regulators**: Automotive industry regulators overseeing the ethical use of AI in car recommendations.
- **Copyright and Image Usage Regulators**: Regulations related to the legal use of car images in the system's database and user uploads, ensuring compliance with copyright laws.

2.3 Functional Requirements

- **Req01**: The system shall allow users to upload photos of cars.
- **Req02**: The system shall allow users to upload car images in common image formats (e.g., JPEG, PNG).
- **Req03:** The system shall support a maximum image size of 50MB.
- **Req04**: When a user uploads a car image, the system shall automatically identify and display the car's make and possibly the model.
- **Req05**: After car model identification, the system shall automatically initiate a search for visually similar cars. The search shall be completed within 5 seconds.
- **Req06**: The system shall display a list of visually similar cars to the identified car model
- **Req07:** For each car in the similar car listing, the system shall display key details such as model name, manufacturer, and representative image.
- **Req08**: The system shall allow users to filter the similar car listing by specifying a desired price range, or by filtering for certain car makes / models only.
- **Req09**: When a user selects a car from the similar car listing, the system shall display a detailed page with more information about the car (e.g., specifications, links to reviews, link to website).

- **Req10**: The system shall provide users with the option to save their search history and preferences for future use.
- **Req11:** When a user uploads a low-quality image, the system shall notify the user and suggest uploading a higher-quality image.
- **Req12:** The system shall allow users to compare multiple car models side by side based on their specifications.
- **Req13**: The system shall provide users with the option to share their search results via social media or email.
- **Req14**: The system shall offer a feedback mechanism for users to report inaccuracies in car model identification or similarity matching.
- **Req15:** If the user provides an image that is not a car, the system shall display the error message: "ImageError: Please upload a picture of a car."
- **Req16:** If the uploaded car image is not in the system's database, the system shall display the error message: "SearchError: This model cannot be identified."
- **Req17:** If the filtering search results in no possible matches, the system shall display the error message: "FilterError: There are no matches to the current search."
- **Req18:** If the uploaded image is not in format JPEG or PNG, the system shall display the error message: "FormatError: Image file not supported. Please upload the image as JPEG or PNG".
- **Req19:** If the uploaded image only contains a part of the car or there is an object obstructing the view of a part of the car, the image shall display the error: "RecognitionError: Picture of the car is missing crucial elements for recognition. Please upload a clear picture of the car."
- **Req20:** If the image recognition is not completed within 5 seconds, the system shall display the error message: "TimeOutError: Search is taking too long. Please try again."
- **Req21:** If the uploaded image exceeds 50MB, the system shall display the error message: "MemoryError: Image exceeds maximum allowed image size. Please upload an image of less than 50 MB."

2.4 Non-Functional Requirements

- **NfReq01 (Performance):** When a user uploads a car image, the System shall identify the car model within 5 seconds.
- **NfReq02 (Performance)**: When initiated, the System shall display the list of similar cars within 5 seconds.
- **NfReq03 (Attributes)**: The System shall have a user interface that is intuitive and easy to navigate for users with varying levels of technical expertise.
- **NfReq04 (Attributes, External Interfaces):** The System shall be accessible and fully functional on both desktop and mobile web browsers.

- **NfReq05 (External Interfaces):** The System shall support image uploads in JPEG and PNG formats.
- **NfReq06 (Constraints)**: The System shall be designed to handle a growing number of users and car image database without significant performance degradation.
- **NfReq07 (Attributes, Performance)**: The System shall achieve a car model identification accuracy of at least 75% on a standard test dataset.
- **NfReq08 (Attributes):** The System's codebase shall be well-structured and documented to facilitate future maintenance and updates.
- **NfReq09 (Performance):** The system shall handle concurrent image uploads from multiple users without significant delay.
- **NfReq10 (Attributes):** The system shall provide multilingual support to cater to a diverse user base.
- **NfReq11 (Attributes):** The system shall ensure that user data, including uploaded images and search history, is stored securely.

2.5 AI-related requirements

- **Req03**: When a user uploads a car image, the system shall automatically identify and display the car's make and possibly the model.
    - **Implementation**: Utilize a pre-trained convolutional neural network (CNN) or Transformer-based Image model fine-tuned on a diverse dataset of cars. The model's latent space can then be utilized to create a “fingerprint” of each car image, in turn yielding a catalogue of fingerprints of car similarity according to the trained model.
- **Req04**: After car model identification, the system shall automatically initiate a search for visually similar cars.
    - **Implementation**: The newly uploaded image is passed through the network to derive the make and model, but more importantly the latent space of this newly image is extracted, and a similarity search is triggered for the latent space of the newly image and the image catalogue already present.
- **Req05**: The system shall display a list of visually similar cars to the identified car model
    - **Implementation**: Through the model’s latent space we can find similar looking cars, based on the latent space of the model, we can then proceed to rank them based on e.g. Cosine-similarity, and yield e.g. the top 5 most similar looking cars.
- **Req13**: The system shall offer a feedback mechanism for users to report inaccuracies in car model identification or similarity matching.
    - **Implementation**: Not technically AI related, but still relevant, gathering feedback via a Thumbs-Up/Thumbs-Down rating system that allows to further analyse the feedback and improve the training and similarity matching process down-the-line.

- **NfReq01 (Performance):** When a user uploads a car image, the System shall identify the car model within 5 seconds.
    - **Justification**: According to user-studies, 5 seconds is already on the brink of having users question the functionality of the system, but at the same time most AI and specifically deep-learning models are resource intensive, hence we'll likely need those 5 seconds to come up with a good result, otherwise the model might need to be smaller and likely deliver less good results overall.
- **NfReq07 (Attributes, Performance)**: The System shall achieve a car model identification accuracy of at least 75% on a standard test dataset.
    - **Justification**: 75% is in fact not a lot, and by using the metric accuracy we additionally don't account for the balanced-ness of the results, so this should be the lowest results achieved otherwise we'd need to question the overall functionality of the system.

3. Use Cases Descriptions

Use case: Identify Car	
ID	UC1
Description	The system identifies the make and model of a car from an uploaded image.
Actors	User, Image Classification Model
Stakeholders:	Users, Car Dealerships, Car Marketplaces
Pre-Conditions	The user has uploaded a clear image of a car.
Success end condition:	The car's make and model are correctly identified.
Failure end condition:	The car cannot be identified.

Main Success Scenario		**Linked UCs**
1	User uploads a car image.	SUC5
2	System processes the image.	SUC1, SUC2
3	Image Classification Model identifies the car.	SUC4
4	System displays the car's make and model.	
Alternative Scenarios		
1	The image is of low quality.	
	System notifies the user to upload a higher-quality image.	
Exception Scenario		
1	The image is not of a car.	
	System displays an error message.	

UC1 is implemented. The system has an identification accuracy of ~85%.

Use case: Car Suggestions	
ID	UC2
Description	The system suggests visually similar cars based on the identified car.
Actors	User, Similarity Matching Model
Stakeholders:	Users, Car Enthusiasts
Pre-Conditions	A car has been identified.
Success end condition:	A list of similar cars is displayed.
Failure end condition:	No similar cars are found.

Main Success Scenario		**Linked UCs**
1	User views the identified car details.	
2	System generates a list of similar cars.	SUC1, SUC2, SUC3
3	System displays the list to the user.	
Alternative Scenarios		
1	User filters the list by price range.	
	System updates the list based on filters.	
Exception Scenario		
1	No similar cars are found.	
	System displays a message indicating no matches.	

UC2 is implemented.

Use case: Search Filtering		
ID	UC3	
Description	The system allows users to filter search results based on preferences.	
Actors	User	
Stakeholders:	Users, Car Dealerships	
Pre-Conditions	A list of similar cars is available.	
Success end condition:	The list is filtered according to user preferences.	
Failure end condition:	No cars match the filters.	

Main Success Scenario		**Linked UCs**
1	User views the list of similar cars.	SUC1, SUC2, SUC3
2	User applies filters (e.g., price range).	
3	System updates the list based on filters.	
Alternative Scenarios		
	User removes a filter.	
	System updates the list accordingly.	
Exception Scenario		
	No cars match the filters.	
	System displays a message indicating no matches.	

UC3 is not implemented.

Use case: Car Comparison		
ID	UC4	
Description	The system allows users to compare multiple car models side by side.	
Actors	User	
Stakeholders:	Users, Car Enthusiasts	
Pre-Conditions	Multiple cars are selected for comparison.	
Success end condition:	Comparison details are displayed.	
Failure end condition:	Comparison cannot be performed.	

Main Success Scenario		**Linked UCs**
1	User selects cars for comparison.	
2	System generates a comparison.	
3	System displays the comparison to the user.	
Alternative Scenarios		
1	User changes the selection of cars.	
2	System updates the comparison.	
xception Scenario		
1	Comparison cannot be generated.	
2	System displays an error message.	

UC4 is not implemented.

Use case: Input Image Analysis (Non-AI)	
ID	UC5
Description	The system analyzes the uploaded image for quality and format.
Actors	User, System
Stakeholders:	Users
Pre-Conditions	User uploads an image.
Success end condition:	Image is accepted for processing.
Failure end condition:	Image is rejected due to quality or format issues.

Main Success Scenario		**Linked UCs**
1	User uploads an image.	SUC5
2	System checks image quality and format.	
3	System accepts the image for further processing.	SUC1, SUC4
Alternative Scenarios		
	The image is in an unsupported format.	
	System notifies the user to upload a supported format.	
Exception Scenario		
	The image exceeds the size limit.	
	System displays an error message.	

UC5 is partially implemented.

Use case: User Preferences and History Management	
ID	UC6
Description	The system allows users to manage their search history and preferences, including saving and sharing search results, and providing feedback on system performance.
Actors	User, System
Stakeholders:	Users, System Administrators
Pre-Conditions	The user is logged into the system.
Success end condition:	User preferences and history are saved, and feedback is recorded.
Failure end condition:	Preferences or history cannot be saved.

Main Success Scenario		**Linked UCs**
1	User logs into the system.	
2	User performs a search and views results.	
3	User saves the search history and preferences.	
3	User shares search results via social media or email.	
4	User provides feedback on the search results and system performance.	
Alternative Scenarios		
	User modifies their saved preferences.	
	System updates the preferences accordingly.	
Exception Scenario		
	System fails to save preferences or history.	
	System notifies the user of the failure.	

UC6 is partially implemented.

Use case: System Scalability and Performance		
ID	UC7	
Description	The system is designed to handle a growing number of users and concurrent image uploads without significant performance degradation.	
Actors	System Administrators, Users	
Stakeholders:	System Administrators, Users	
Pre-Conditions	The system is operational.	
Success end condition:	The system handles increased load efficiently.	
Failure end condition:	The system experiences performance degradation.	

Main Success Scenario		**Linked UCs**
1	Multiple users upload images concurrently.	
2	System processes each upload without significant delay.	
3	System maintains performance as the user base grows.	
Alternative Scenarios		
	System administrators monitor system performance.	
	System administrators optimize performance based on monitoring data.	
Exception Scenario		
	System performance degrades under heavy load.	
	System administrators are notified and take corrective action.	

UC7 cannot be implemented in this stage of the application.

Use case: Multilingual Support and Documentation		
ID	UC8	
Description	The system provides multilingual support and maintains a well-structured and documented codebase to facilitate future maintenance and updates.	
Actors	Developers, Users	
Stakeholders:	Developers, Users, System Administrators	
Pre-Conditions	The system is under development or maintenance.	
Success end condition:	The system supports multiple languages, and the codebase is well-documented.	
Failure end condition:	Multilingual support or documentation is incomplete.	

Main Success Scenario		**Linked UCs**
1	Developers implement multilingual support features.	
2	Developers document the codebase thoroughly.	
3	Users access the system in their preferred language.	
Alternative Scenarios		
	Developers update documentation as new features are added.	
	System administrators review and approve documentation.	
Exception Scenario		
	Documentation is found to be incomplete during a review.	
	Developers are notified to update the documentation.	

UC8 is partially implemented (codebase).

Use case: Data Security and Privacy		
ID	UC9	
Description	The system ensures that user data, including uploaded images and search history, is stored securely	
Actors	System Administrators, Users	
Stakeholders:	Users, System Administrators, Regulators	
Pre-Conditions	Users are interacting with the system.	
Success end condition:	User data is securely stored.	
Failure end condition:	User data is compromised.	

Main Success Scenario		**Linked UCs**
1	Users upload images and save search history.	
2	System securely stores the data.	
3	System administrators monitor data security.	
Alternative Scenarios		
	Users request to delete their data.	
	System securely deletes the data.	
Exception Scenario		
	A data breach is detected.	
	System administrators take immediate action to secure the data and notify affected users.	

UC9 is implemented.

Supporting Use case: Image Fingerprint		
ID	SUC1	
Description	The system generates a unique "fingerprint" for an image based on its features.	
Actors	Image Processing Model	
Stakeholders:	Users, Car Dealerships, Car Marketplaces	
Pre-Conditions	An image is available for processing	
Success end condition:	A fingerprint is generated.	
Failure end condition:	Fingerprint generation fails.	

Main Success Scenario		
1	User uploads a car image.	
2	System processes the image.	
3	Image Classification Model generates a fingerprint.	
Alternative Scenarios		
	The image is of low quality.	
	System notifies the user to upload a higher-quality image.	
Exception Scenario		
	The image is not of a car.	
	System displays an error message.	

SUC1 is implemented.

Supporting Use case: Latent Feature Extractor	
ID	SUC2
Description	The system extracts latent features from the car image.
Actors	Image Processing Model
Stakeholders:	Users, AI Engineers, Car Enthusiasts
Pre-Conditions	An image has been uploaded.
Success end condition:	Latent features are extracted.
Failure end condition:	Features cannot be extracted.

Main Success Scenario	
1	User uploads a car image.
2	System processes the image.
3	Image Classification Model extracts latent features.
Alternative Scenarios	
	The image is of low quality.
	System notifies the user to upload a higher-quality image.
Exception Scenario	
	The image is not of a car.
	System displays an error message.

SUC2 is implemented.

Supporting Use case: Cosine Similarity	
ID	SUC3
Description	The system calculates cosine similarity between car images.
Actors	Similarity Matching Model
Stakeholders:	Users, System Administrators
Pre-Conditions	Fingerprints of car images are available.
Success end condition:	Similarity is calculated
Failure end condition:	Similarity cannot be calculated.

Main Success Scenario	
1	User uploads a car image.
2	System generates a fingerprint.
3	System calculates cosine similarity with other fingerprints.
Alternative Scenarios	
	The image is of low quality.
	System notifies the user to upload a higher-quality image.
Exception Scenario	
	The image is not of a car.
	System displays an error message.

SUC3 is implemented.

Supporting Use case: Image Classification	
ID	SUC4
Description	The system classifies the car image to identify the make and model.
Actors	Image Classification Model
Stakeholders:	Users, System Administrators
Pre-Conditions	An image has been uploaded.
Success end condition:	The car is correctly classified.
Failure end condition:	The car cannot be classified.

Main Success Scenario	
1	User uploads a car image.
2	System processes the image.
3	Image Classification Model classifies the car.
Alternative Scenarios	
	The image is of low quality.
	System notifies the user to upload a higher-quality image.
Exception Scenario	
	The image is not of a car.
	System displays an error message.

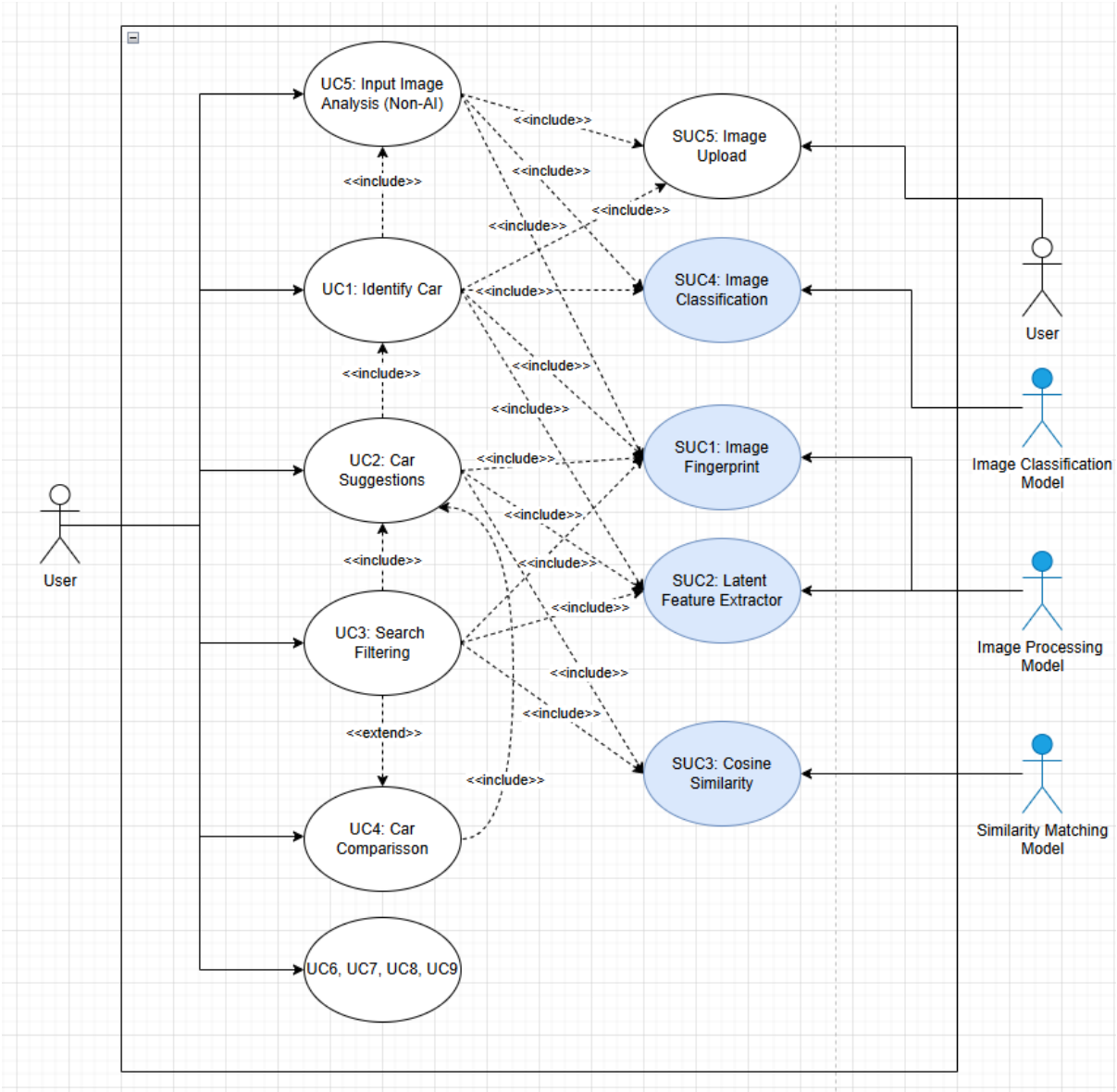
SUC4 is implemented.

Supporting Use case: Image Upload	
ID	SUC5
Description	The system handles the upload of car images by users.
Actors	User, System
Stakeholders:	Users, System Administrators
Pre-Conditions	User is on the upload page.
Success end condition:	The image is successfully uploaded.
Failure end condition:	The image upload fails.

Main Success Scenario	
1	User selects an image to upload.
2	System receives the image.
3	System confirms the upload.
4	Image is moved forward to the classification process
Alternative Scenarios	
	The image is in an unsupported format.
	System notifies the user to upload a supported format.
Exception Scenario	
	The image exceeds the size limit or an upload error occurs.
	System displays an error message.

SUC5 is implemented.

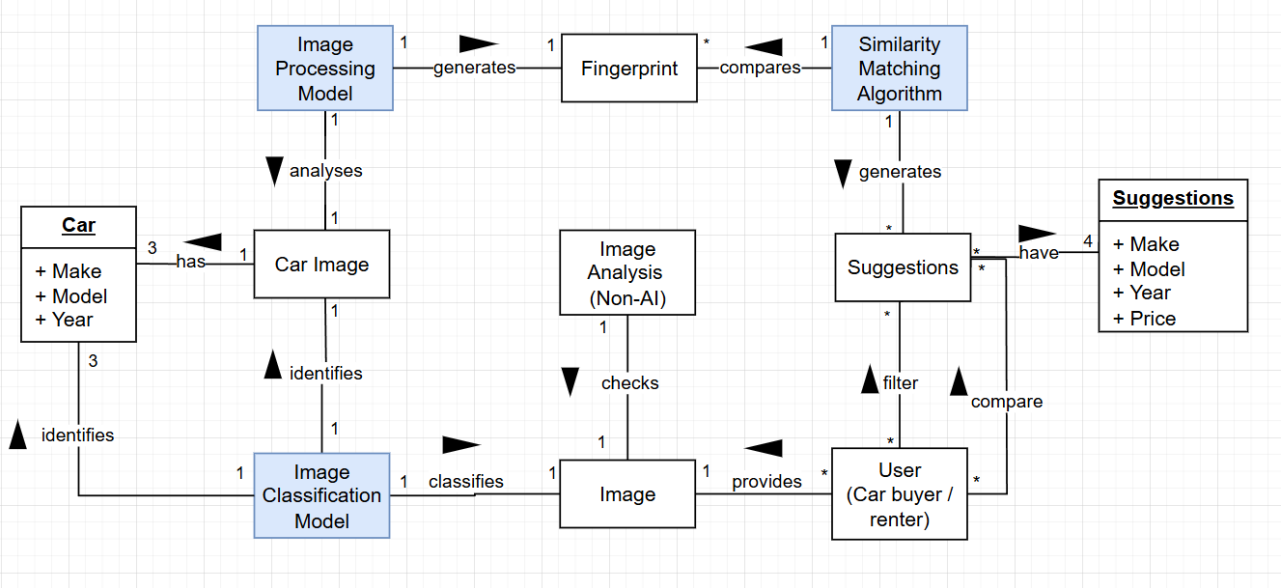
4. Use Cases Diagram



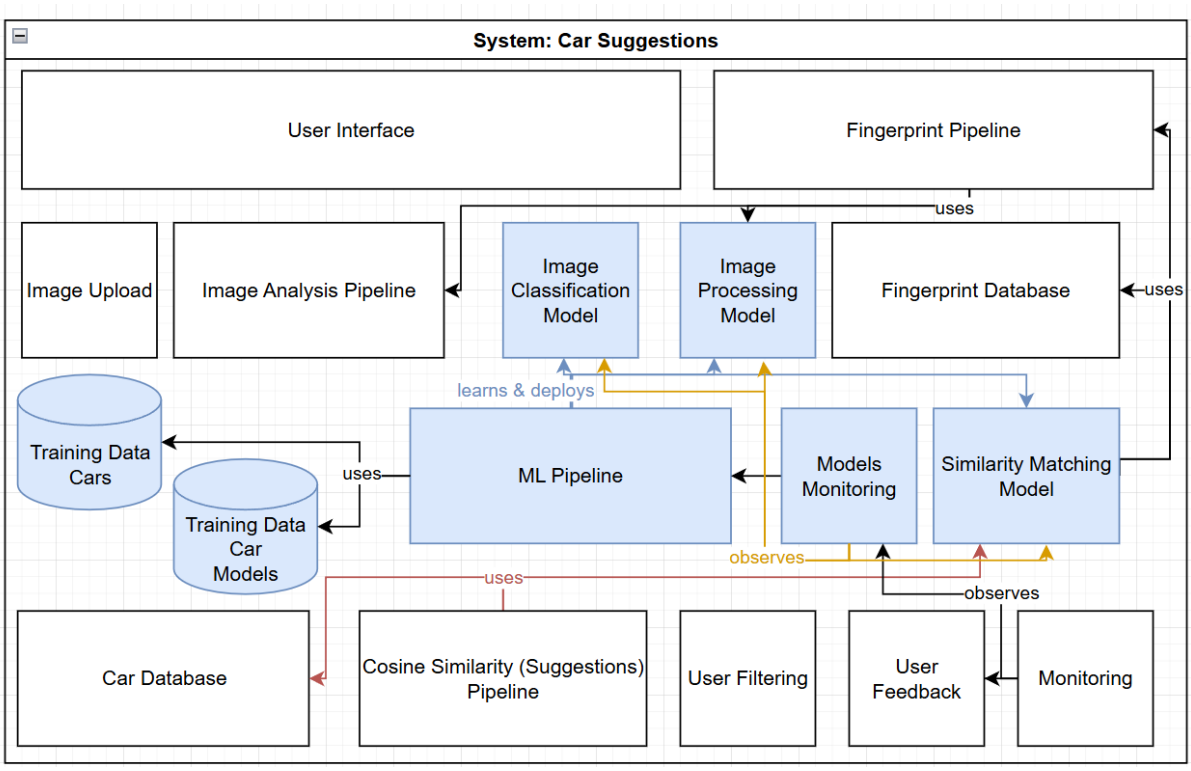
5. Traceability Matrix

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	SUC1	SUC2	SUC3	SUC4	SUC5
Req01	x				x								x	x
Req02					x									x
Req03					x									x
Req04	x									x	x		x	
Req05	x	x								x	x	x		
Req06		x	x	x						x	x	x		
Req07		x		x										
Req08			x											
Req09			x	x										
Req10						x			x					
Req11	x													x
Req12			x	x										
Req13						x			x					
Req14	x					x						x	x	
Req15	x												x	
Req16	x												x	
Req17		x	x											
Req18	x				x									x
Req19	x				x								x	
Req20							x						x	x
Req21	x				x									x
NfReq01	x						x			x	x		x	
NfReq02		x					x					x		
NfReq03	x	x	x	x			x							
NfReq04	x	x	x	x										
NfReq05	x													x
NfReq06							x							
NfReq07	x												x	
NfReq08								x						
NfReq09	x						x							x
NfReq10		x	x	x			x	x						
NfReq11							x		x					x

6. Domain Model



7. Architecture Diagram



8. Components Description

User Interface: The application's front-end. User-friendly platform where they can upload photos (Req01), view a list of similar car suggestions (Req06), filer suggestions based on user preferences (Req08), make a selection of a car and view detailed information (Req09), provide feedback (Req14), and see error messages (Req15-21).

Fingerprint Pipeline: Image processing and analysis workflow that generates a "fingerprint" of the car and compares it with a database to find similar cars. (Req05, Req06, Req07).

Image Upload: Allows users to upload images into the system. (Req01)

Image Analysis Pipeline: Processing steps to evaluate the uploaded image. Checks for supported formats (Req02), maximum image size (Req03).

Image Classification Model: A ML model that identifies first if the uploaded image is a car (Req01, Req15), and then identifies the car's make and model (Req04).

Image Processing Model: Model that extracts the feature space of the Image Classification Model and generates a fingerprint of the car. (Req04)

Fingerprint Database: Database that contains the "fingerprints" of car images for comparison (Req05).

Training Data Cars: Dataset of images used to train car vs no car classification. (Req15)

Training Data Car Models: Dataset of car images used to train the model to identify car models. (Req04)

ML Pipeline: Manages the flow of data through the machine learning models.

Models Monitoring: Tracks the performance of the deployed models; evaluates their accuracy.

Similarity Matching Model: Model that compares the generated "fingerprint" of the car with the fingerprint database to generate suggestions (Req05).

Car Database: Database containing detailed information about car models. (Req07)

Cosine Similarity (Suggestions) Pipeline: Ranks the similar cars based on the fingerprints of the car. (Req06)

User Filtering: Allows the user to filter the car listing by their preferences. (Req08)

User Feedback: Allows the user to report inaccuracies of the car identification and suggestions (Req14).

Monitoring: Evaluates system performance, usage and user interactions.

9. Design Questions & Answers

Design Questions

1. How will the system handle variations in image quality and lighting conditions to ensure accurate car model identification?

The system will preprocess images to normalize overall conditions. Additionally, the model will be trained on a diverse dataset including various lighting and quality scenarios to improve robustness.

2. What strategies will be employed to ensure the scalability of the system as the database of car images and user base grows?

The system will employ a service-oriented architecture, allowing independent scaling of components like image processing and database management.

3. How will the system manage and update the "fingerprint database" to include new car models and updates to existing ones?

The database will be designed with versioning to track updates and additions. Automated scripts will periodically update the database with new models.

4. What measures will be taken to ensure the security and privacy of user-uploaded images and associated data?

User images will be transferred via HTTPS, hence be encrypted during the transmission process. Access controls will be implemented to restrict data access.

5. How will the system provide real-time feedback to users regarding the status of image processing and potential delays?

The system will provide visual progress indicators and notifications to inform users about processing stages.

6. What techniques will be used to optimize the performance of the machine learning model to meet the 5-second identification requirement?

Techniques such as model quantization and pruning will be used to reduce computational requirements.

7. How will the system handle and recover from errors such as unsupported image formats or incomplete car images?

The system will implement robust error handling mechanisms to detect and notify users of issues like unsupported formats or incomplete images. Automated retries and fallback mechanisms will be in place to recover from transient errors.

8. What user interface design principles will be applied to ensure an intuitive and user-friendly experience across different devices?

The UI will follow responsive design principles to ensure usability across devices.

9. How will the system integrate feedback from users to continuously improve the accuracy of car model identification and similarity matching?

A feedback loop will be established where users can report inaccuracies. This feedback will be analysed to identify patterns and improve the model through continuous retraining and updates.

10. What strategies will be employed to ensure the system is accessible and functional across different devices and platforms?

Cross-platform compatibility will be ensured through thorough testing on various devices and browsers to provide a consistent user experience.