

Commands with description, which type of users can invoke them and user story assigned to them (for personal reference).

Actor	Story	Command	Description
Admin	Accept or refuse a comment	ApproveComment()	Admin can approve or refuse a comment
Admin	Model, Mark, Fuel,...	Model() Mark() Fuel() Class() Gearbox()	Adding, editing and deleting these entities
Admin	Block and (De)activate users	Block() Activate() Deactivate()	Managing users
Admin	Register new agent/company	Agent()	Register new agent
Admin	Permissions	Permissions()	Change permissions
Admin	Manage user requests for registration	AcceptRegistration()	Accept or refuse users registration request
Agent	Statistics	Statistics()	Get statistics of kilometrage, grades, etc
Unregistered	Register	Register()	User registers
User	Rent cars	RentRequest()	Request to rent a car from a user
User	Rent bundle of cars	RentBundle()	Request to rent multiple cars as a bundle
User	Get requests	GetRequests()	Get all requests from a user
User	Leave a mark	Grade()	Grade
User	Cancel a request	Cancel()	Cancel a request before payment
User/ Agent	Leave a comment	Comment()	Leave a comment
User/Agent	Login	Login()	User logs in the app
User/Agent	Accept request	AcceptRent()	Accept request or bundle
User/Agent	Refuse request	RefuseRent()	Refuse request
User/Agent	Create an ad	Ad()	Create an add, user 3 agent unlimited
User/Agent	Add occupation	Occupation()	Add times when car is unavailable to rent and refuse all requests for

			that period
User/Agent	Get my ads/cars	MyCArs()	Get ads from one user and all its comments and marks
User/Agent	Add Report	Report()	Add report for each rented car If kilometrage is higher than expected, automatically sent bill to user
User/Agent	Pricelist	PriceList()	Price for day, discount, cwd, kilometrage
User/Agent	Message	Message()	Message each other
User/Unregistered	Search	Search()	Search for cars
User/Unregistered	Advanced search	AdvancedSearch()	Advanced search
User/Unregistered	Car details	GetAd()	Get details of an ad

Application divided by business capabilities:

- Car management service
 - adding cars
 - adding reports
 - adding availability
 - managing entities (model, mark, fuel, class..)
 - comments
- Ads management service:
 - new ad
 - searching cars
 - adding price list
 - comment
- Admin service (managing users)
 - adding agents
 - managing roles and permissions
 - messaging between users or agent user
- Security service
 - login
 - register
 - change password
- Request management service
 - accepting refusing
 - all requests
 - creating new
 - bundles
- grades

Operations and in which service are they going to be in.

Operation	Service	Entities needed for execution	Preconditions	Post-conditions	Return value
Ad(user id, date from, date to, location string, car car)	Ad	User, Car, Ad	User exists Car exists or is created	Car is created	adId
AdvancedSearch(...)	Ad	Ad, Car, Model, Mark, Class, Fuel, Gearbox, Occupied, PriceList		At least one car was found	Collection<Ad>
Comment(car id, user id, comment string)	Car	Ad, Car, Comment	Car exists User exists	Comment is associated to car	CommentId
GetAd(adId)	Ad	Ad, Car, Model, Mark, Class, Fuel, Gearbox, Report, Comment, Occupied, PriceList	Ad exists		Ad
PriceList(price per day, cwd, price per kilometrage, discount)	Ad	Pricelist		Price list was created	PricelistId
Search(date from, date to, location)	Ad	Ad, Car		At least one ad was found	Collection<Ad>
AcceptRegistration(user id, accept/refuse)	Admin	User	User exists and is disabled	User is either enabled or deleted	userId
Agent(name, surname, company, address, bussinesId)	Admin	Agent	Company does not already exist	Agent was created	agentId
ApproveComment(comment id, approve or deny)	Admin	Comment	Comment exists	Comment was accepted or refused	CommentId
Block(user id) Activate(user	Admin	User	User exists	User blocked/(de)ac	UserId

id) Deactivate(user id)				tivated	
Message(sender id, receiver id, text)	Admin	Message, User	Sender exists Receiver exists	Message was associated to users	/
Permissions(user id)	Admin	User, Permission	User exists	Permission granted or taken	UserId
Model(name) Mark(name) Fuel(type) Class(type) Gearbox(type)	Car	Model, Mark, Fuel, Class, Gearbox	Does not already exist	Created	Id of the entity
MyCars(user id)	Car	Car, Comment, User	User exists		Collection<Car>
Occupation(date from, date to, car id)	Car	Car, Occupied	Car exists	Refused all requests in that period	CarId
Report(kilometrage, comment, car id)	Car	Car, Report	Car exists		CarId
Statistics(user id)	Car	Car	User exists		Dto with statistics
AcceptRent(request id)	Request	RentRequest	Request exists	Rent is payed in 12 hours All other requests are canceled	AdId
Cancel(request id)	Request	RentRequest	Request exists	Request is canceled	RequestId
GetRequests(user id)	Request	RentRequest, User	User exists		Collection<Requests>
Grade(car id)	Request	Car, Grade	Car exists	Cars grade Is recalculated	CarId
RefuseRent(request id)	Request	RentRequest	Request exists	Request is refused	RequestId
RentBundle(list of ads, user id)	Request	Ad, User, RentRequest	Ads exist User exists	Requests are in pending state	UserId
RentRequest(ad id, user id)	Request	Ad, User, RentRequest	Ad exists User exists	Request is in pending state	RequestId
Login(username, password)	Security	User	User with associated username is registered	Authentication token is created and assigned to	AuthToken

				user	
Register(name, surname, username, password)	Security	User	Username does not exist	User is created and disabled	userId

Napomena:

Inicijalno su Ad service i Car service bili jedan servis, međutim htjeli smo da nam pretraga bude highly available pošto je to jedini dio aplikacije kojem mogu pristupiti i neregistrovani korisnici, a i Car service je postajao relativno velik, pa smo pretragu izdvojili u odvojen servis. Dodatno smo u servis za pretragu ubacili neke funkcionalnosti koje su vezane za reklame za rentiranje, pošto su izgledale relativno bliske. Međutim, nakon sastavljanja ove tabele shvatili smo da operacije iz Ad service-a koriste mnoge entitete iz Car service-a, tako da možda dođe do prebacivanja nekih metoda u Car service, da se izbjegne učestala komunikacija između ova dva servisa.