Software Engineering - Analysis
Project: E-Scooter Rental Service

Kendra Birringer (1229372)
Nader Cacace (1208115)
Steffen Hanzlik (1207417)
Marco Peluso (1228849)
Svetozar Stojanovic (1262287)

Frankfurt University of Applied Sciences
Faculty 2

February 15th, 2020

Contents

1 Introduction

In this project, we took the role of a start-up company building an E-Scooter rental business. For analyzing this project, we were encouraged to use the agile method *Scrum* [1] as a framework.

The main objective was to develop a high-level software architecture for the E-Scooter rental business. To be able to do this we had to gain a clear understanding of the system's requirements and to build a model which fulfilled these. We used the *Unified Modeling Language (UML)* [2] in conjunction with the modeling tool *MagicDraw* [3] to achieve this.

We also had to build user interface prototypes for relevant parts of the functionality. User interface prototypes help developers and customers to "better understand how the system will work and whether it would meet the requirements." [4] Concerning the customer, presenting her/him the prototypes, it is very important that she/he understands that these prototypes are not the finished product but only illustrative examples to demonstrate functionality. We used *Axure RP 9* [5] to create our prototypes.

For our project, the artifacts consist of diagrams and documentation describing the software functionality. The quality of these artifacts was ensured by a proper definition of "Done".

2 Definition of "Done"

The definition of "Done" (DoD) is part of the Scrum metrics. All members of the Scrum Team must have a shared understanding of what it means when the work is complete, to ensure transparency. [6] It is a (check-)list of items which need to be validated to consider a backlog item being "Done". DoD is defined by the development organization to make sure that the results of multiple teams can be integrated into a releasable product. [7]

For our project, the result needed to match the following definition of "Done":

- ☐ Description of the requirement in form of a Use Case
- ☐ Categorization of requirement (functional/non-functional, client/server)
- ☐ Business value of the corresponding functionality
- ☐ Effort estimation for the implementation of the requirement

☐ For UI related functions: UI prototype

☐ UML Diagrams

- Use case diagram
- Activity diagram
- Class diagram
- Sequence diagram

☐ Detailed documentation (e.g. table) about who worked on the item and what has been done during the sprint.

☐ Overall quality of the documentation meets general industry standards.

☐ The results have been reviewed and accepted by another member of the team (tester). It needs to be documented who has performed the review.

3 High-level Software Architecture

As you can see in Figure 1, our team decided to concentrate on analyzing the Software for the Mobile Application first. We figured, this was the main part of our system, and since we had a limited amount of time, we wanted to create the most important part first. Desktop-/Laptop-Clients could be addressed later on.

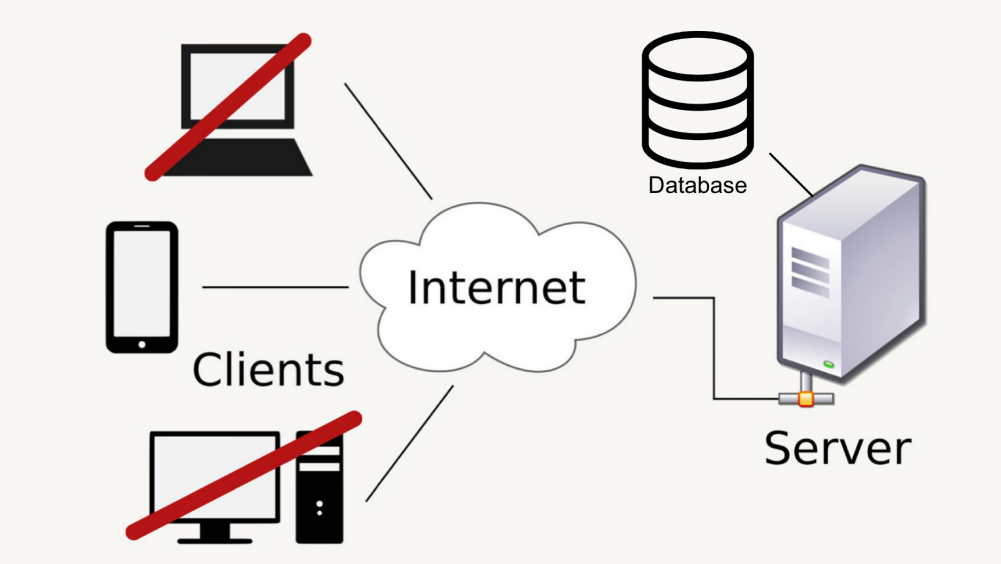


Figure 1: High-level software architecture

Source: https://commons.wikimedia.org/wiki/File:Client-server-model.svg (modified by Kendra Birringer)

4 Backlog Items

Backlog items are part of a product backlog. This is an ordered list of requirements which have to be satisfied according to the DoD for the product. The purpose of a backlog item is to take a requirement description, in form of a use case description, as input and build an analysis model of our software capability as result.

We used the *Volere Requirements Specification Template* [8] also known as *Volere Snow Cards* as a template to create a list of backlog items using Google Sheets where we collected all of our backlog items.

This requirement matrix is shown below.

ID	Name	Requirement Type	Associated Roles	Event/ use case #	Description	Rationale	Source	Fit Criterion	Customer Satisfaction	Customer Dissatisfaction	Dependencies	Conflicts	Supporting Materials	Priority	Estimation	Status	User Story
1	Registration	Functional	Customer	Register	Customers need to register to get personal information, payment method etc.	Customers must register to use the E-Scooter rental service	Steffen Hanzlik, Marco Peluso, Kendra Birringer	Fits if the customer can register successfully	5	5	Registration success	None		High	8	Done	As a new customer, if I want to use the service, I must be able to register and create a new account.
2	Log-in	Functional	Customer	Manage Account	Customers must log-in if they want to rent an E-Scooter	Customers must log-in to use the E-Scooter rental service	Steffen Hanzlik, Marco Peluso, Kendra Birringer	Fits if the customer can log-in	4	4	Register	None		High	5	Done	As a customer, if I want to use an E-Scooter, I must be able to log-in into my account.
3	Log-out	Functional	Customer	Manage Account	Customers must have the possibility to log-out	Customers should be able to log-out from our service	Steffen Hanzlik, Marco Peluso, Kendra Birringer	Fits if the customer can log out	3	3	Register, Log-in	None		High	2	Done	As a customer, if I finished the ride and do not need the E-Scooter anymore, I must be able to log-out of my account.
4	Change Account Details	Functional	Customer	Manage Account	Change personal information or payment	Customers should be able to change their personal information	Steffen Hanzlik, Marco Peluso, Kendra Birringer	Fits if the customer can change account details successfully	3	5	Register, Log-in	None		Medium	3	Done	As a customer I would like to be able to change my account information if something has changed, like personal information or payment method.
5	Delete Account	Functional	Customer, Admin	Manage Account	Customers and Admins can delete a customer's account	Customers should be able to delete their account if they do not want to use the E-Scooter Service any further. Also, if the customer is not following the Terms and Conditions, the Admins also should be able to remove the customer's account	Svetozar Stojanovic	Fits if the customer's account no longer exists	3	3	Register, Log-in	None		Medium	2	Done	As a customer, I want to be able to delete my account if I do not want to use the service anymore. / As an Admin I want to be able to delete a customer's account, if she/ he violates the Terms and Conditions.
6	Choose Subscription Plan	Functional	Customer	Manage Account	Three membership types: Gold, Silver and Bronze. Customers without a membership get their prices calculated per km or min/ h while the customers with a membership pay more affordable prices on a monthly/ yearly basis.	Customers should be able to choose to pay on a monthly or yearly basis	Nader Cacace, Svetozar Stojanovic	Fits if the customer has applied to a membership successfully	3	3	Register, Log-in, Device supporting GPS	None		Low	3	Done	As a customer, I want to be able to choose the Subscribtion plan which fits my needs.
7	Change Subscription Plan	Functional	Customer	Manage Account	Customers can change their membership type whenever they want	To give Customers the possibility to adapt to their needs as soon and flowed as possible	Nader Cacace	Fits if the customer can change the membership type easily and without any disruption	5	4	Choose Membership	None		Low	1	Done	As a customer I would like to change my membership type according to my needs.

ID	Name	Requirement Type	Associated Roles	Event/ use case #	Description	Rationale	Source	Fit Criterion	Customer Satisfaction	Customer Dissatisfaction	Dependencies	Conflicts	Supporting Materials	Priority	Estimation	Status	User Story
8	Cancel Subscription	Functional	Customer	Manage Account	Customers can cancel the membership type without deleting their whole account	If a Customer knows she/ he will not use the service for a certain period of time, she/ he can cancel the membership without deleting the whole account.	Nader Cacace	Fits if the Customer can cancel her/ his membership when the service is not needed for this customer but nevertheless could keep the account	3	4	Choose Membership	None		Low	2	Done	As a customer I like to cancel my subscription plan to save money if I know that I will not use this service in the next time.
9	Check statistics (customer)	Functional	Customer	Manage Account	Customers can check their own statistics	The customer can see the total price for all rides or only for a specific ride or check the total distance covered etc.	Nader Cacace	Fits if the customer gets a nice and easy to understand overview of all rides he/she ever done with the service	3	2	Register, Log-in	None		Low	5	Done	As a customer I would like to see my own statistics, like how much money spent, distance covered etc., so I can possibly adjust my subscription plan.
10	Day Pass System	Functional	Customer	Manage Account	Customers can pay for a daily pass	If a customer knows, she/ he will use our Service extensively throughout a day	Svetozar Stojanovic, Nader Cacace, Marco Peluso	Fits if the customer can purchase a Day Pass in the app	2	1	Register, Log-in	None		Low	3	Done	As a customer, if I know I need an E-Scooter most of the day, I would like to purchase a Day Pass.
11	Purchase Gift Cards	Functional	Customer	Manage Account	Customers can purchase gift cards and send them to other people via E-Mail	If the customer wants to recommend the service to other potential customers and pay for them, they can do that through gift cards	Svetozar Stojanovic, Nader Cacace, Marco Peluso	Fits if the customer is able to purchase gift cards and send them to other people via E-Mail	3	1	Register, Log-in	None		Low	8	Done	As a customer I would like to purchase gift cards and be able to send them to other people.
12	Redeem Promo Codes, Gift Cards	Functional	Customer	Manage Account	Customers can redeem promo codes/ vouchers/ gift cards in the app	Promo Codes and vouchers should attract new customers to our service	Svetozar Stojanovic, Nader Cacace, Marco Peluso	Fits if the customer can redeem promo codes/ vouchers/ gift cards in the app	5	5	Register, Log-in	None		High	5	Done	As a customer, if I have a promo code/ voucher/ gift card, I would like to be able to redeem it in the App.
13	Check-out E-Scooter	Functional	Customer	Check-out	Check-out reperesents the state in which the customer scans the QR Code of the E-Scooter and starts the ride.	Customers should be able to rent an available E-Scooter whenever they need	Steffen Hanzlik, Marco Peluso, Kendra Birringer	Fits if the E-Scooter checks-out and the customer can ride it	5	5	Register, Log-in, valid payment	None		High	5	Done	As a customer, if I want to ride an E-Scooter, I scan the QR Code of a Scooter and tap the "Check-out E-Scooter" button in the App.
14	Reserve an E-Scooter	Functional	Customer	Check-out	Customers should have the possibility to pre-book an E-Scooter for later or for another day	To make sure that customers can take an E-Scooter from a specific place at a specific time	Nader Cacace	Fits if the customer can reserve an E-Scooter and so she/ he can be sure that it is ready and charged at the given place and time	3	3	Register, Log-in	None		Medium	13	Done	As a customer, I would like to be able to reserve a Scooter so that I can take the ride at a specific time and I want to be sure, that the Scooter is ready at the given place and time.

ID	Name	Requirement Type	Associated Roles	Event/ use case #	Description	Rationale	Source	Fit Criterion	Customer Satisfaction	Customer Dissatisfaction	Dependencies	Conflicts	Supporting Materials	Priority	Estimation	Status	User Story
15	Show allowed use area for E-Scooter	Functional	Customer	Check-out	Shows a map of the area where customers are allowed to ride the E-Scooters	To prevent that the customer does not drive too far away from the "legal area". To give customers a better experience, the E-Scooter should not slow down if they leave the "legal area" so that they are possible to check-in the E-Scooter	Nader Cacace	Fits if the customer can see a map with the exact area where the E-Scooter can be driven and checked in after the ride	2	5	Check-out E-Scooter	None		Medium	5	Done	As a customer I need to know the area in which I am allowed to ride and return the Scooter.
16	Check-in E-Scooter	Functional	Customer	Check-in	Check-in represents the state in which the customer has finished the ride.	Customers should be able to finish the E-Scooter usage to pay just for what she/ he needs	Steffen Hanzlik, Marco Peluso, Kendra Birringer	Fits if the E-Scooter checks-in and the amount of costs are fixed	5	5	Register, Log-in, valid payment	None		High	5	Done	As a customer, when I finished the ride and do not need the E-Scooter anymore, I tap on the "Check-in E-Scooter" button in the App.
17	Cancel E-Scooter reservation	Functional	Customer	Check-in	Customers should have the chance to cancel a reserved E-Scooter if they do not need it anymore	To bring this E-Scooter back to the "available" mode so that other customers could rent this E-Scooter	Nader Cacace	Fits if the reserved E-Scooter is available after cancellation	5	5	Reserve E-Scooter	None		Medium	1	Done	As a customer I like to cancel my E-Scooter reservation if I do not need the Scooter anymore.
18	Warn user about approaching border of allowed use area	Functional	Customer	Check-in	Send a warning to the customer if she/ he is leaving the area where the E-Scooters can be checked-in	Used to prevent the Customers having a bad experience with the check-in process because she/ he will not be able to check-in the E-Scooter at any point of the city	Nader Cacace	Fits if the customer gets a push notification before she/ he leaves the "legal area"	2	5	Check-out E-Scooter	Lost GPS could be an issue		Medium	8	Done	As a customer I would like to get a warning, if I almost leave the area where I can bing back and check-in the Scooter.
19	Get GPS Location of the Customer	Functional	Customer	Find an E-Scooter on the map	Displays the customer's position on the map using GPS	Customers should be able to see their relative position to the nearby available E-Scooters	Svetozar Stojanovic	Fits if the customer can see her/his position on the map in the App	5	4	Register, Log-in, GPS enabled device	No GPS location provided		High	8	Done	As a customer I would like to see my position on a map and all available Scooters nearby.
20	Get GPS Location of the E-Scooter	Functional	Customer, Maintenanceman	Find an E-Scooter on the map	Displays all available E-Scooters on the map using a built-in map and GPS locator	Customers should see nearby E-Scooters on the map	Svetozar Stojanovic	Fits if the customer sees the available E-Scooters on the map in the App	5	2	Register, Log-in	None		High	13	Done	As a customer, I need to see the GPS Location of an available Scooter to find it and rent it. / As a Maintenanceman, I need to see the GPS Location of a Scooter to collect them and recharge them.
21	Download offline maps	Functional	Customer	Find an E-Scooter on the map	Customers can download an offline map of their area	In order to speed up the usage of the App, the map of the customer's area can be downloaded	Svetozar Stojanovic	Fits if the customer has downloaded the map for the offline use	5	2	Register, Log-in	None		Low	8	Done	As a customer I would like to be able to download an offline map, to speed up the App or to save data volume and batterylife of my phone.
22	Generate QR Code	Functional	Admin	Scan QR Code	QR Code is generated from an external source	For each new E-Scooter there is a need for a newly generated QR Code	Svetozar Stojanovic	Fits if the QR Code is generated as an unique code	3	3	External QR Code Generator	None		Medium	2	Done	As an Admin I want that every E-Scooter has an unique QR Code.
23	Display QR Code	Functional	Admin	Scan QR Code	QR Code which is unique for each E-Scooter is being displayed on each of them	QR Code should be displayed on each E-Scooter in order to connect the E-Scooter with the App	Svetozar Stojanovic	Fits if the QR Code is displayed correctly on the E-Scooter	4	3	An E-Scooter with a charged battery	E-Scooter's battery is dead		Medium	2	Done	As an Admin I want that an QR Code, which is unique for each E-Scooter, is displayed on each of them.
24	Scan QR Code	Functional	Admin, Customer	Scan QR Code	QR Code is being scanned in order to book an E-Scooter	Customer scans the E-Scooter so the ride can start	Svetozar Stojanovic	Fits if the customer has scanned the QR Code with her/ his E-Scooter App successfully	5	5	Register, Log-in, Find an E-Scooter on a map	Customer's device does not have a camera, E-Scooter App does not scan the code correctly		High		Done	As a customer, if I would like to ride an E-Scooter, I need to be able to scan the QR Code of the Scooter via the App, so that I can check it out and start the ride.

ID	Name	Requirement Type	Associated Roles	Event/ use case #	Description	Rationale	Source	Fit Criterion	Customer Satisfaction	Customer Dissatisfaction	Dependencies	Conflicts	Supporting Materials	Priority	Estimation	Status	User Story
25	Push Notifications	Non - Functional	Admin	Get Notifications	Send a push notification to the customer's device	Customers should be able to get notifications	Svetozar Stojanovic	Fits if the notification is sent successfully	3	3	Register, Log-in	None		Medium	5	Done	As an Admin I would like to be able to send notifications to the customer's device via the App.
26	Show Driving Rules	Functional	Admin	Get Notifications	Driving rules are displayed to the customer's device	Before driving an E-Scooter, each newbie to driving an E-Scooter should be presented with driving rules (i.e. on which lane to drive, speed limit when driving next to pedestrians, basic traffic rules, etc.)	Svetozar Stojanovic	Fits if the brief driving rules are displayed successfully to the display of the customer before the ride	2	4	Register, Log-in	None	Some driving rules basic handout/ rule list	Low	1	Done	As an Admin, I want to show all new customers, before they ride an E-Scooter for the first time, the driving rules. This includes the allowed lanes, speed limit, basic traffic rules, etc.
27	Show Safety Risks	Functional	Admin, Customer	Get Notifications	Show safety risks from driving the E-Scooter inappropriately are displayed to the customer	Safety risks when driving under influence (drugs, alcohol, certain medications), medical conditions, etc.	Svetozar Stojanovic	Fits if all safety risks/ health hazards associated with the inappropriate usage of the E-Scooter are dispalyed to the customer	2	4	Register, Log-in	None	Some kind of basic safety risks when misused	Low	1	Done	As an Admin I want to show the safety risks of diving an E-Scooter to the customers.
28	Show Legal Information	Functional	Admin, Customer	Get Notifications	Legal constraints are displayed to the customer	If the Customer tries to misuse the E-Scooter, the risks should be presented to her/ his device's display	Svetozar Stojanovic	Fits if the Customer sees the legal risks when misusing the E-Scooter in the App	3	3	Register, Log-in	None	Some kind of a basic legal notice	Low	1	Done	As an Admin I want to show the legal constraints of using an E-Scooter. If a customer misuses a Scooter, she/ he will be presented with the risks on her/ his display.
29	Send E-Mail Newsletter	Functional	Customer	Get Notifications	Customer receives an E-Mail with news about the App, company or special offers etc.	Customers should be able to learn about the news	Svetozar Stojanovic	Fits if the customer has received an E-Mail containing the news about the software and company, etc.	2	2	Register, Log-in	None		Low	1	Done	As a customer , if I agree, I would like to receive the newsletter to get the latest news about offers, promo codes, etc.
30	Wallet to manage account balance	Functional	Customer	Pay the Service	Wallet is an in-app virtual money storage system designed for the convinient use by the customer	A wallet is needed for managing the customer's money balance for the ride	Svetozar Stojanovic, Nader Cacace, Marco Peluso	Fits if the wallet functionality exists and the customer can send her/ his money to the wallet	5	3	Register, Log-in	None		High	8	Done	As a customer, I would like to have an easy way to pay for the service with a nice overview about my account balance.
31	Refund Customer to wallet	Functional	Admin	Pay the Service	Admin refunds the ride to the in-app wallet of the customer	Admins may need to refund his ride directly to the wallet if the customer cannot drive the specific scooter	Svetozar Stojanovic, Nader Cacace, Marco Peluso	Fits if the customer has her/ his money refunded to the wallet	5	5	Register, Log-in	Refund to the original payment method		High	8	Done	As an Admin, I would like to have the possibility to refund the price for a ride if something went wrong for the customer.
32	Refund Customer to original payment method	Functional	Admin	Pay the Service	Service refund if the Scooter does not start. The service fee is then sent to the original payment method used for paying the fee.	Customer should get her/ his money back if the Scooter does not drive	Svetozar Stojanovic, Nader Cacace, Marco Peluso	Fits if the refund is sent to the original payment method	4	1	Register, Log-in, Check-out, Check-in	Refund to the Customer wallet		High	13	Done	As an Admin I need to pay back the money to the customer if a Scooter does not drive.
33	Send Payment Confirmation	Functional	Payment Service	Payment	Payment confirmation is sent to the rental service's bank account notifying them that the customer has paid his service charges	Customers should be able to pay in their favorite payment service	Steffen Hanzlik, Marco Peluso, Kendra Birringer	Fits if the payment was successful	4	4	Register, Log-in	None		High	3	Done	As Payment Service I will send a notification to the rental service's bank account, to inform that a customer has paid her/ his service charges.

ID	Name	Requirement Type	Associated Roles	Event/ use case #	Description	Rationale	Source	Fit Criterion	Customer Satisfaction	Customer Dissatisfaction	Dependencies	Conflicts	Supporting Materials	Priority	Estimation	Status	User Story
34	Ask for feedback	Functional	Admin, Customer	Give Feedback	Customer is prompted with a menu asking for a feedback in the App	Customers should be asked to give a feedback because many forget or are not bothered to give a feedback	Svetozar Stojanovic	Fits if the the UI panel with the 'Ask for a feedback' question is displayed and the customer can give an immediate feedback	2	4	Register, Log-in	None		Medium	2	Done	As an Admin I would like to be able to ask customers to give feedback. / As a customer I would like to be able to give feedback.
35	Give feedback	Functional	Customer	Give Feedback	Customers should be able to give feedback to the company if they have suggestions to improve the service	Customers should have the possibility to give good or bad feedback to the company who provide the service	Nader Cacace	Fits if the customer is able to send feedback to the company	4	4	Register, Log-in	None		Medium	8	Done	As a customer, I need the possibility to give feedback if I have any suggestions to share.
36	Analyze customer feedback	Functional	Admin	Analyze Customer Feedback	The company should be able to collect customer satisfaction and suggestions for improvements	The company can improve the service with the suggestions from the customers and keep their satisfaction on a high level	Nader Cacace	Fits if the Admin can see the suggestions from the customers	5	5	Register, Log-in	None		High	8	Done	As an Admin I like to analyze the customer feedback to improve our service.
37	Check numbers of customers	Functional	Admin	Check statistics	Shows the company the number of customers	To check the amount, age, gender etc. of customers. This could help to better target the ordains for advertising	Nader Cacace	Fits if the company can check the number of customers in a given period	3	3	Register, Log-in	None		Meidum	1	Done	As an Admin I would like to see the amount of customers, their ages, genders etc. The statistics could help to better target the ordains for advertising.
38	Check overall revenue	Functional	Admin	Check statistics	The company should be able to check the earnings per day/ month/ quarter or year	To plan future investments	Nader Cacace	Fits if the company can check their earnings	5	5	Register, Log-in	None		High	1	Done	As an Admin I would like to see all earnings over a certain period of time.
39	Check revenue per customer	Functional	Admin	Check statistics	Shows revenue per customer in a given period or all in one	To help the company to build the right pricing model for them and the customers	Nader Cacace	Fits if the company is able to check revenue per customer. Maybe per specific ordinance or age groups	4	5	Register, Log-in	None		Medium	1	Done	As an Admin I want to check the revenue of specific/ all customers in a given period of time or all in one. This helps the company to build the right pricing models for the customers
40	Report problems	Functional	Customer, Maintenanceman/ Admin	Report problems	Send information about specific problems from an E-Scooter or the App	To help the Maintenanceman/ Admin to find the problems of an E-Scooter faster and make it easier to fix the relevant E-Scooter or the App	Nader Cacace	Fits if the Maintenanceman/ Admin gets Information about the problem	3	5	Register, Log-in	None		High	8	Done	As a customer, I want to be able to report a problem with an Scooter or the App if something is not working. / As a Maintenanceman/ Admin I want to get information if something is wrong with a Scooter or the App.
41	Show E-Scooter status	Functional	Maintenanceman	Maintain the scooters	Shows the current status of a specific E-Scooter	To check in which state the different E-Scooters currently are (Batterylife cycles, total distances, last checkup, etc.)	Nader Cacace	Fits if the Maintenanceman can check all parameters of any E-Scooter	5	5	Database with information about all E-Scooters	None		High	1	Done	As a Maintenanceman I would like to see the current status of a specific Scooter, like batterylife, total distances, last check-up etc.
42	Show Battery status	Fuctional	Admin, Customer	Maintain the scooters	Check battery status of any E-Scooter	To find all empty E-Scooters to recharge them	Nader Cacace	Fits if the company could check the battery status of every E-Scooter to find the empty ones and charge them	5	5	Software on the E-Scooter	None		High	2	Done	As a customer I would like to know if the battery of the Scooter is sufficient so that I can decide if it has enought power to bring me to my destination. / As an Admin, I want to know the battery level of the Scooters to know which ones have to be collected by the Maintenanceman and recharge them.

ID	Name	Requirement Type	Associated Roles	Event/ use case #	Description	Rationale	Source	Fit Criterion	Customer Satisfaction	Customer Dissatisfaction	Dependencies	Conflicts	Supporting Materials	Priority	Estimation	Status	User Story
43	Add E-Scooter to the System	Functional	Maintenanceman	Add E-Scooter to System	If a new E-Scooter arrives, the stock has to be updated and the number of avalible E-Scooters has to be increased	To keep an overview about available E-Scooters	Nader Cacace	Fits if the maintenance database got an update (+1)	2	2	QR Code Generation	None		High	3	Done	As a Maintenanceman I like to add a new Scooters to the system to administrate the amount of available Scooter for the rental service.
44	Delete E-Scooter from System	Functional	Admin	Delete E-Scooter from System	If an E-Scooter is not longer in use, the stock should be updated (decreasing the number of avalible E-Scooters)	To keep an overview about available E-Scooters	Nader Cacace	Fits if the maintenance database got an update (-1)	2	2		None		Medium	3	Done	As an Admin, I need to delete an old or broken Scooter from the system to maintain the system and clean up the database.
45	Easy to use	Non-Functional	Customer	App handling	The complete UI design should suggest an easy and logical operation of the App	Every customer should be able to understand every part of the App and be able to use it	Nader Cacace Svetozar Stojanovic	Fits if the customer uses the app with full understanding	5	1	App installed	None		Medium		Done	As a customer I would like to have an App that is easy to understand and easy to use
46	Reliable	Non-Functional	Customer	App handling	The service should work anywhere and anytime	In order to keep the customer's satisfaction high, the App should work as reliable as possible	Nader Cacace	Fits if the customer can be sure that the service works in a comprehensive way	5	5	Register, Log-in	None		High		Done	As a customer I want an Application which is as reliable as possible.
47	App Language	Non-Functional	Customer	Choose Language	Languages we want to display for the customer in our App: English, German	The language variety is important to attract new Csustomers for our Service, the more language options we have, the more Csutomer we would attract	Steffen Hanzlik	Fits if the customer sees the language on the Application Screen	5	5	Register, Log-in	None		Medium	5	Done	As a customer I would like to be able to change the App language.
48	Get the amount of E-Scooters in System	Functional	Maintainanceman	Maintain the scooters	Shows the amount of E-Scooters in the system	To check if there is a need for new E-Scooters and if so to purchase them	Nader Cacace	Fits if the Maintenancemean/ Admin can see the number of avalible E-Scooters (In Use, Charging, under repair, etc.)	4	4	Database with information about all E-Scooters	None		Medium	2	Done	As a Maintenanceman/ Admin I would like to see the amount of all E-Scooters in the system and to check if there is a need for new E-Scooters

5 Week One

In the first week a few of our team members rented an E-Scooter from the company "Lime" to gain a better understanding of the process of renting an E-Scooter. That really helped us to find additional requirements for our E-Scooter rental service project. As we already mentioned before, we collected those requirements in a Google Sheets document. That way we could easily collaborate on them, even if we sometimes could not meet at the same location. We also started to talk about the requirements' estimations, satisfactions conditions, priorities, and fit criteria.

The estimation of the effort required to complete the backlog items, was difficult for our team, since we were all inexperienced in that regard. That's why we decided to play "Planning Poker", which is "[...] a consensus-based estimating technique. Agile teams around the world use Planning Poker to estimate their product backlogs. Planning Poker can be used with story points, ideal days, or any other estimating unit." [9] This proved to be very helpful and was also a lot of fun for our team.

We mostly spent the rest of the week with the collection of more requirements for the project.

Since our team members had vastly different schedules, finding dates for further meetings turned out to be very difficult. That lead us to the decision to have regular weekly instead of daily meetings and whenever there would occur any problems which needed to be discussed, via "Discord". Discord is an "All-in-one voice and text chat [...] that's free, secure, and works on both your desktop and phone."[10]

5.1 Division of Work: Week One

K. Birringer	N. Cacace	S. Hanzlik	M. Peluso	S. Stojanovic
20%	20%	20%	20%	20%

6 Week Two

Since we decided to use the agile method Scrum for analyzing the E-Scooter rental service project, the team members were assigned the following roles:

- Development Team: Kendra Birringer, Nader Cacace, Steffen Hanzlik, Marco Peluso, Svetozar Stojanovic

  The Development Team was responsible for modeling all neccessary UML diagrams and sketching UI prototypes.

- Scrum Master: Kendra Birringer

  In addition to her role as a member of the development team, Kendra also took on the role of the Scrum Master. She was responsible for the organization of the whole team: she organized and moderated the team meetings and wrote the protocols.

  Another task was to check and correct the spelling, grammar and contents of everything that was written.

- Tester: Marco Peluso

  Marco also acted as the tester. He mainly reviewed, documented and accepted the resulting artifacts.

In the second week the team discussed each of the collected requirements, and we decided which of them fit and which are not necessary for the software. During this discussion we gathered more requirements.

Then we started to talk about the UML diagrams and built a first use case diagram which was too big and complex and needed some adjustments. So, the task for the Development Team was to simplify the diagram and make it clearer. Our solution for this problem was to build smaller use case diagrams in addition to the big overview, that just display a single use case in each of them (see Appendix). This was done by Kendra, Svetozar and Marco.

Also, Kendra, Marco and Steffen built a main structure for the documentation of the project and started writing the documentation with LaTex.

6.1 Division of Work: Week Two

K. Birringer	N. Cacace	S. Hanzlik	M. Peluso	S. Stojanovic
18%	21%	22%	18%	21%

7 Week Three

In week three the Development Team modified the use case diagram which was too big. They also modeled further use case diagrams which we then discussed, to find out if they needed further adjustments. At the end of this week, we finished the use case diagrams and Kendra finally added the use case documentation to each use case.

Also, after we thought about where it could be necessary, some activity and sequence diagrams were modeled.

Steffen and Kendra started modeling the following activity diagrams:

- Check-in E-Scooter
- Check-out E-Scooter
- Log-in
- Log-out
- Manage Account
- Register

Svetozar and Marco modeled the following activity diagrams:

- Give Feedback
- Pay the Service
- Report problems with the Scooter

Nader worked on the following sequence diagrams:

- Check-in E-Scooter
- Check-out E-Scooter
- Add E-Scooter to System
- Find an E-Scooter on the Map
- Reserve an E-Scooter
- Register

- Report Problems with the Scooter

Marco reviewed Nader's sequence diagrams and did some tweaks and adjustments to them.

Also, Svetozar and Marco worked on the Wallet Management sequence diagram.

Then we started to think about the class diagram and asked ourselves which classes we needed and which relations the different classes could have to each other. Then Steffen and Marco started modeling the class diagram.

Furthermore, the first UI prototype "Start Menu" was built with the software design tool *Axure* [5]. This was mainly done by Svetozar.

7.1 Division of Work: Week Three

K. Birringer	N. Cacace	S. Hanzlik	M. Peluso	S. Stojanovic
19%	12%	25%	19%	25%

8 Week Four

During week four we refined the state of the project by doing a lot of adjustments and modifications to everything we had done so far. We updated the requirements, Kendra checked the spelling and grammar, put the requirements in a proper order and Marco checked what was still missing and controlled and adjusted all of the diagrams which needed further improvements.

On the basis of the backlog items list we finished all UML diagrams and Svetozar finished all UI prototypes (see Appendix). After we heard the lecture, in which Prof. Dr.-Ing Peter Thoma talked about UML diagrams, Steffen noticed, that all our activity diagrams lacked a cancellation mode. Therefore, Steffen and Kendra adjusted all these diagrams and added a cancellation mode to them.

Marco and Kendra finished most of the documentation, so that only the appendices needed to be added.

The goal of our team was, to finish everything until the end of this week, so that in the following week, we would be able to fully concentrate on the presentation of the project.

8.1 Division of Work: Week Four

K. Birringer	N. Cacace	S. Hanzlik	M. Peluso	S. Stojanovic
22%	12%	21%	20%	25%

9 Week Five

In week five, we checked everything again and finished everything. Kendra generated the use case report from MagicDraw. Because the smaller use cases were not included in the report, you can find them in the Appendix.

Also all team meeting protocols were added by Marco and Kendra.

The user interface prototypes were added as screenshots to the appendix to the project's documentation.

Once the documentation was finished, each team member checked it again and made any necessary improvements, corrections and additions.

Then, finally the whole team started to work on preparing the presentation. At first, our presentation was a bit too detailed to fit into the 10 minute time limit, which we found out during our first rehearsal. We trimmed it down bit by bit until we could comfortable perform it under 10 minutes (see Appendix).

9.1 Division of Work: Week Five

K. Birringer	N. Cacace	S. Hanzlik	M. Peluso	S. Stojanovic
23%	11%	20%	30%	16%

10 Conclusion

Despite our team not being able to meet at the same location most of the time, our process did turn out to work quite well. As we already mentioned, we adapted the Scrum framework to our specific needs in terms of meetings, in that we only meet once a week, regularly and used previously mentioned collaboration tools like Discord, Google Sheets and GitHub.

In the middle of the of the project we became a little stuck and could not complete all the sprint items. We had problems with the sequence and activity diagrams. First, we had to do some more research regarding these type of diagrams. This slowed down our workflow and we had to finish a few sprint items a week later, than we had scheduled. Also we had to get familiar with the UI prototyping tools. In the last 3 weeks, we had less problems and worked more to our schedule/planning.

Overall we faced a few problems, but in the end everything was done to the best of our abilities. This experience was really educational for this project and our further projects. The work could be clearly divided, because of the sprint items and everybody always knew what they had to do. Communication between the team members worked very well due to the regular meetings and tools we used.

Finally, this project was one of the biggest projects in our studies so far and we learned a lot. We are very grateful that we had this experience.

11 Appendix

11.1 Use Case Report

E-Scooter App

Use Case Report

Author: Author

Revision: 0.2

No Magic

One Allen Center, 700 Central Expressway
South, Suite 110 Allen, Texas 75013 USA

Date: February 09, 2020

REVISION HISTORY

Revision	Date	Reason For Changes	Author
0.1	<date 1>	<draft>	<Author name>
0.2	<date 2>	<release>	<Author name>

TABLE OF CONTENTS

TABLE OF FIGURES

Introduction

Purpose

<This document provides an overview of use case view>

Scope

<Provide a short description of the system being specified and its purpose, including relevant benefits, objectives, and goals>

Overview

<Describe what the document contains and explain how document is organized>

E-Scooter App

Actor Bank

Actor E-Payment Service

1 UseCase Register

New customers need to download the App and register before they can use the E-Scooter rental service.

- Register is performed by Customer
- Register includes Send an email confirmation

Extension Points

- Subscription: Register

Description

Pre-Condition

Customer has downloaded the App.

Post-Condition

Customer has successfully registered and now has an account.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

Normal

Scenarios

Basic Flow

1. Fill in User Information
2. Confirm User Information

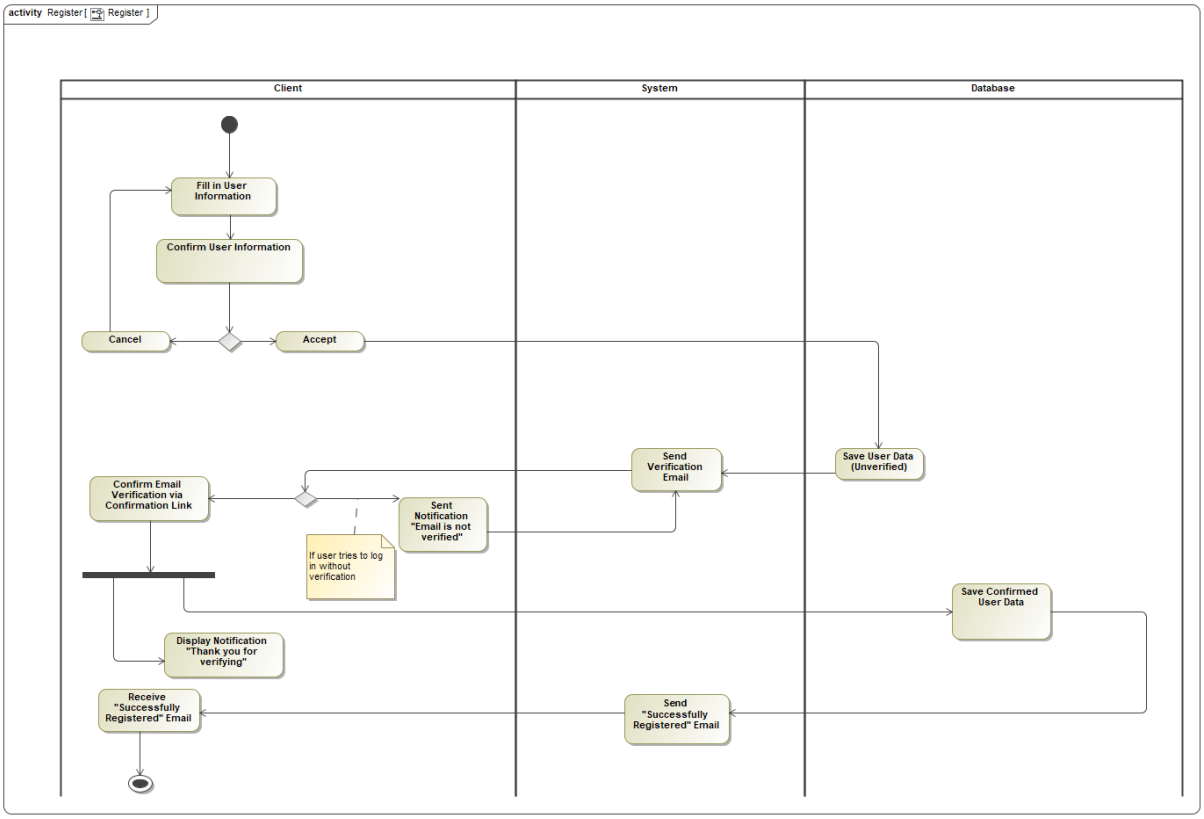


Figure 1. Register

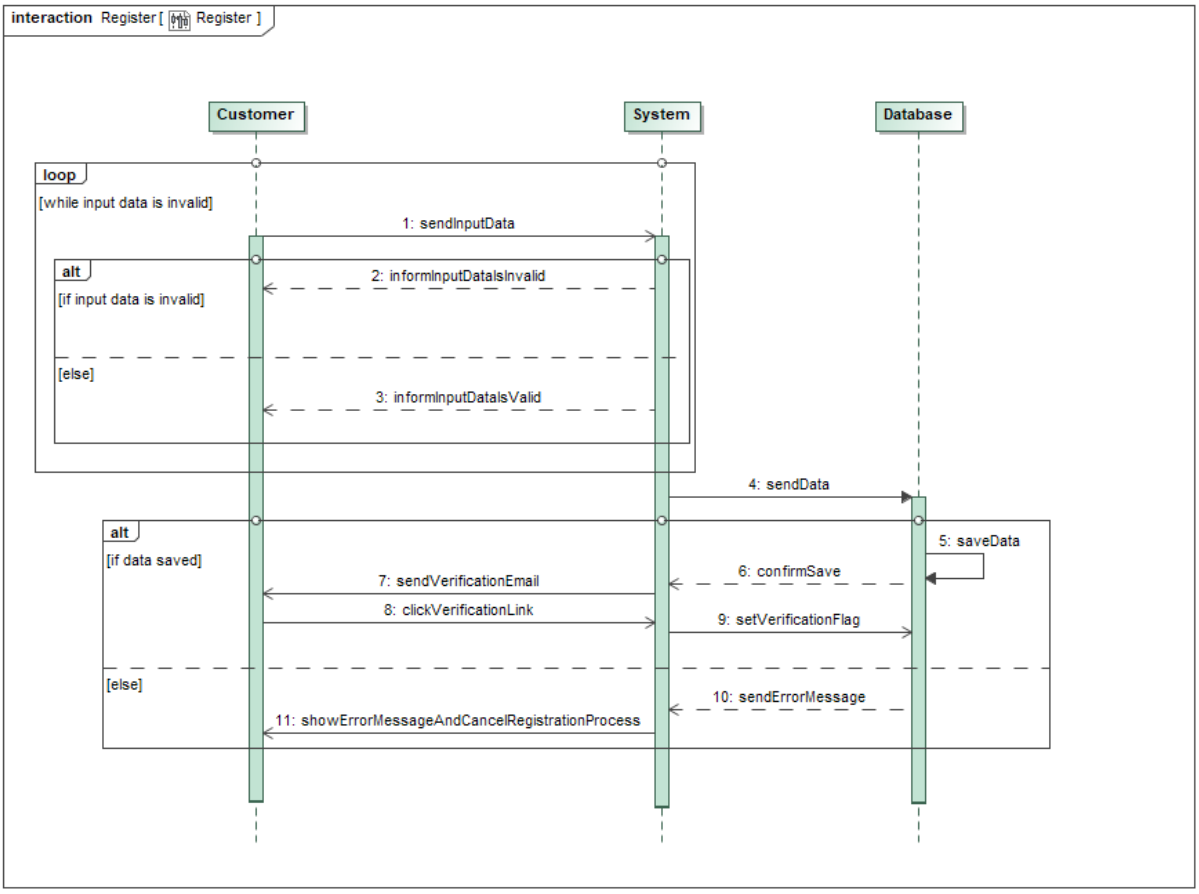


Figure 2. Register

2 UseCase Add E-Scooter to System

If a new E-Scooter arrives, the stock has to be updated and the number of available E-Scooters has to be increased.

- Add E-Scooter to System is performed by Maintenance man

Description

Pre-Condition

New Scooters are shipped to the service/maintenance location.

Post-Condition

New Scooters are registered and ready for its first customer.

Goal

New E-Scooters are properly added to the database and are ready for the first ride.

Complexity

Low Complexity

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

0. Servicer makes a quality check for the fresh E-Scooter.
1. Admin generates new QR-Code for the specific Scooter.
2. Servicer registers the Scooters to the local government registration office, in order to get the plates.
3. Admin registers the E-Scooter to the Database.
4. Servicer places the E-Scooter on the appropriate Scooter stations.

Exceptional Flow

1. Quality check fails.
2. Scooter are returned to the seller.
3. New Scooters are ordered.
4. See the Basic Flow of Events.

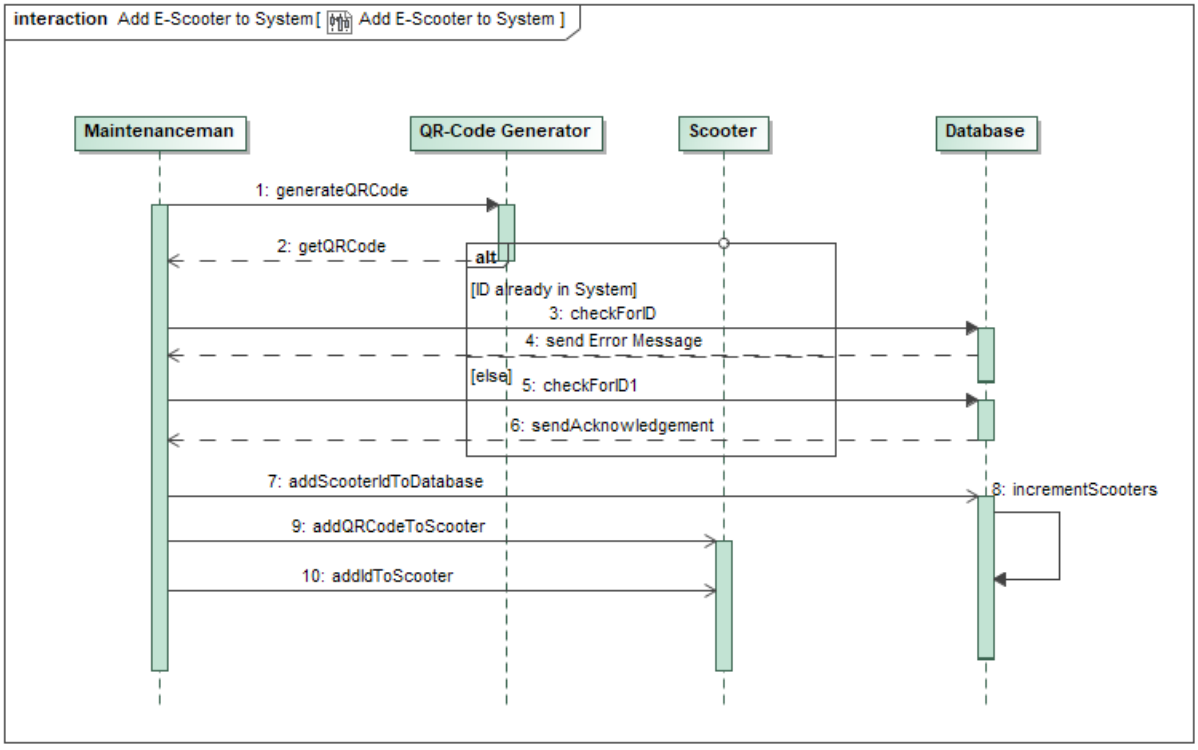


Figure 3. Add E-Scooter to System

3 UseCase Change Bank Credentials

Customers should be able to change their bank credentials.

- Change Bank Credentials extends Manage Account

Description

Pre-Condition

Customer is logged in.

Post-Condition

Customer's credit card credentials are changed.

Goal

Customers can change their credit/debit card information.

Complexity

Average Complexity

Assumption

Credit card information is either already given or it is the first time giving these details.

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

1. Customer logs in.
2. Customer navigates to the "Payment Information/Payment Methods/Credit Debit Card".
3. Customer either enters the information for the first time or she/he changes the credentials.
4. Customer clicks the "Save" button.
5. UI refreshes.

4 UseCase Check drive statics (all users)

Admins can check the drive statistics of all/ specific customers.

- Check drive statics (all users) extends Check Statistics (Admin)

Description

Goal

See overall user's habit data pattern.

Complexity

Average Complexity

Non-Functional Requirements

Improve E-Scooter Placement

Component Complexity

Average Complexity

Priority

Low

5 UseCase Check overall revenue

Shows Admins the earnings per day/ month/ quarter/ overall revenues.

- Check overall revenue extends Check Statistics (Admin)

Description

Pre-Condition

At least multiple customers have driven the E-Scooters.

Post-Condition

Administrator knows the current revenue from the overall customer's usage.

Goal

Knowing the current income situation from the E-Scooters.

Complexity

Average Complexity

Non-Functional Requirements

Improve Revenue (Business type non-functional)

Component Complexity

Average Complexity

Priority

Below Normal

Scenarios

Basic Flow

1. Administrator checks the sum of all transactions in the database.

6 UseCase Check Statistics (Admin)

Admins can check statistics like total amount of customers, overall revenue or revenue per customer.

- Check Statistics (Admin) is performed by Administrator
- Check Statistics (Admin) is performed by Administrator

Extension Points

- Check drive statistics (all users) : Check Statistics (Admin)
- Check revenue: Check Statistics (Admin)

Description

Pre-Condition

At least enough statistically significant number of the E-Scooter users.

Post-Condition

Overall pattern of the user's drive habit data is collected and ready for the analysis.

Goal

Know the customer's drive habit in the city in order to improve the placement of the E-Scooters.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

Below Normal

7 UseCase Check usage statistics (Customer)

Customers can check their own statistics, like total price for all rides or total distance covered, etc.

- Check usage statistics (Customer) extends Manage Account

Description

Pre-Condition

Customer has logged in.

Post-Condition

Customer sees her/his drive data on a diagram.

Goal

Presents to the customer his/her drive habit data in the past days/weeks/months/ years

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

Normal

8 UseCase Check-out E-Scooter

Check-out represents the state in which the customer has scanned the QR Code of an E-Scooter in order to ride on it.

- Check-out E-Scooter is performed by Customer
- Check-out E-Scooter is performed by Customer
- Check-out E-Scooter includes Scan QR Code
- Check-out E-Scooter includes Show E-Scooter Battery Status

Description

Pre-Condition

Customer has opened the App and has logged in.

Post-Condition

Customer has successfully checked out his/her E-Scooter.

Complexity

High Complexity

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

1. Open Application
2. Scan QR- Code

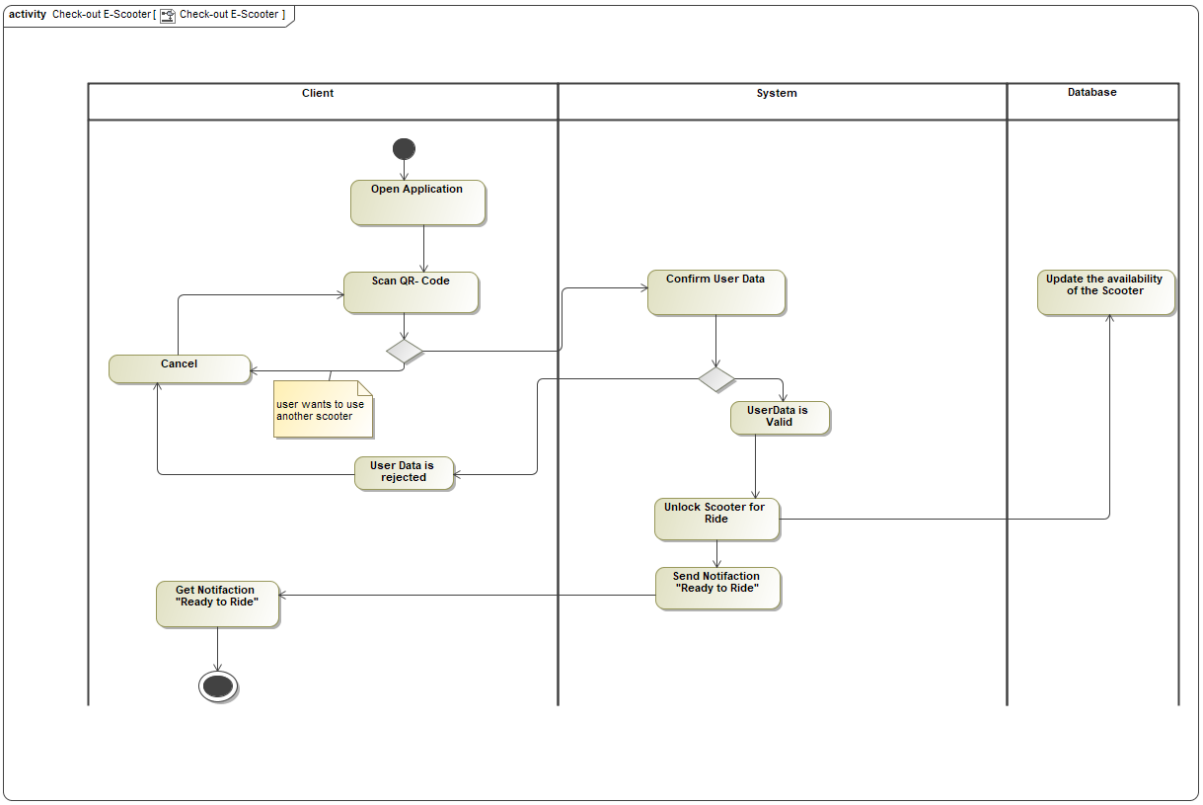


Figure 4. Check-out E-Scooter

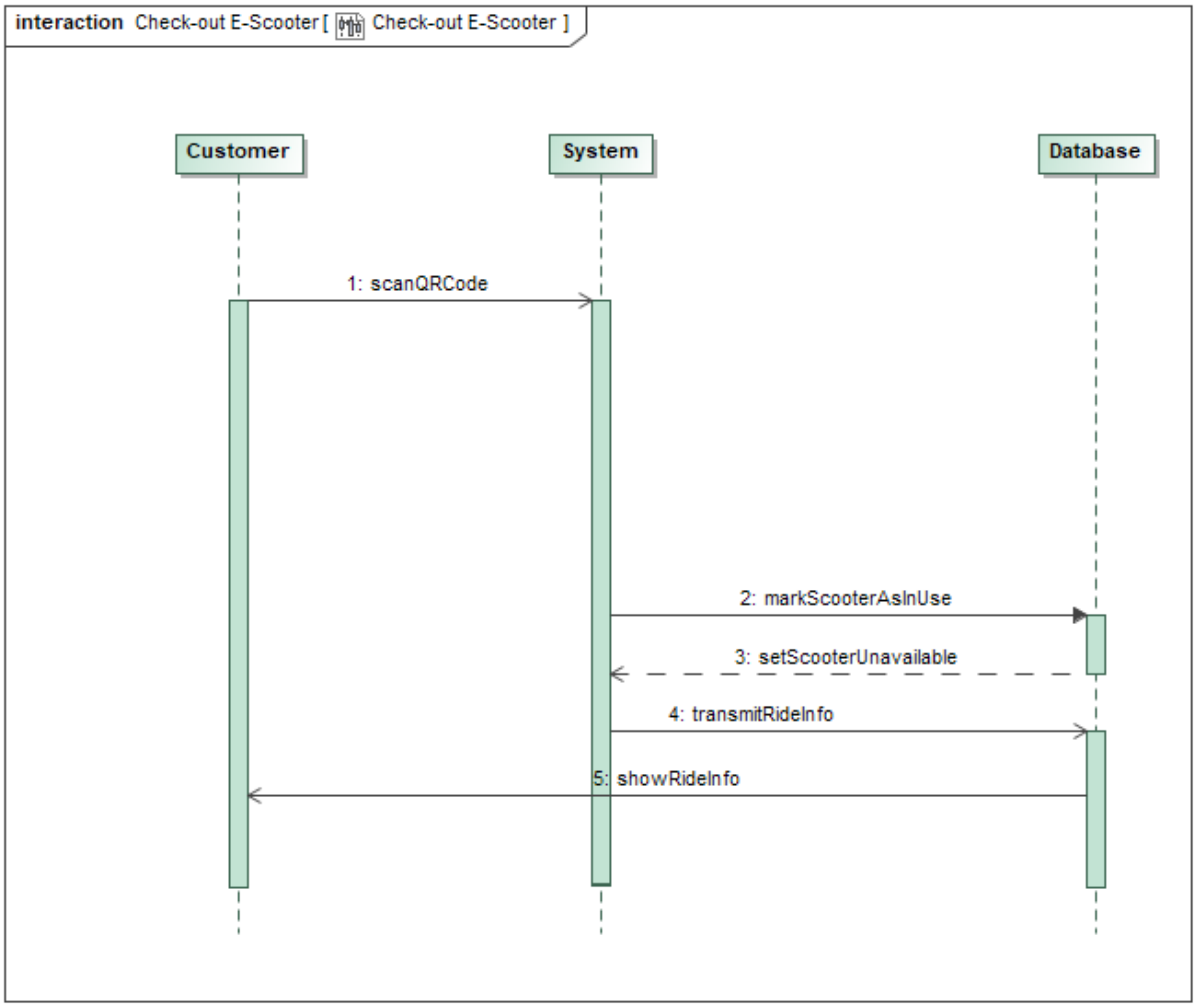


Figure 5. Check-out E-Scooter

9 UseCase Check-in E-Scooter

Check-in represents the state in which the customer has finished the usage of an E-Scooter.

- Check-in E-Scooter is performed by Customer
- Check-in E-Scooter includes Pay the service

Description

Pre-Condition

Customer has already checked out an E-Scooter.

Post-Condition

Customer has successfully terminated the ride and parked the Scooter.

Goal

Make Customer be able to return/park the E-Scooter to an appropriate location within the drive area limits.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

Normal

Scenarios

Basic Flow

1. Display Application Start screen
2. Set Status "Ride Finished"
3. Get Status "Ride Finished"
4. Sent Notification to the user, so it is sure that she/he finished the ride and must pay for it
5. Get Notification to confirm that the user has finished the ride and must pay for it

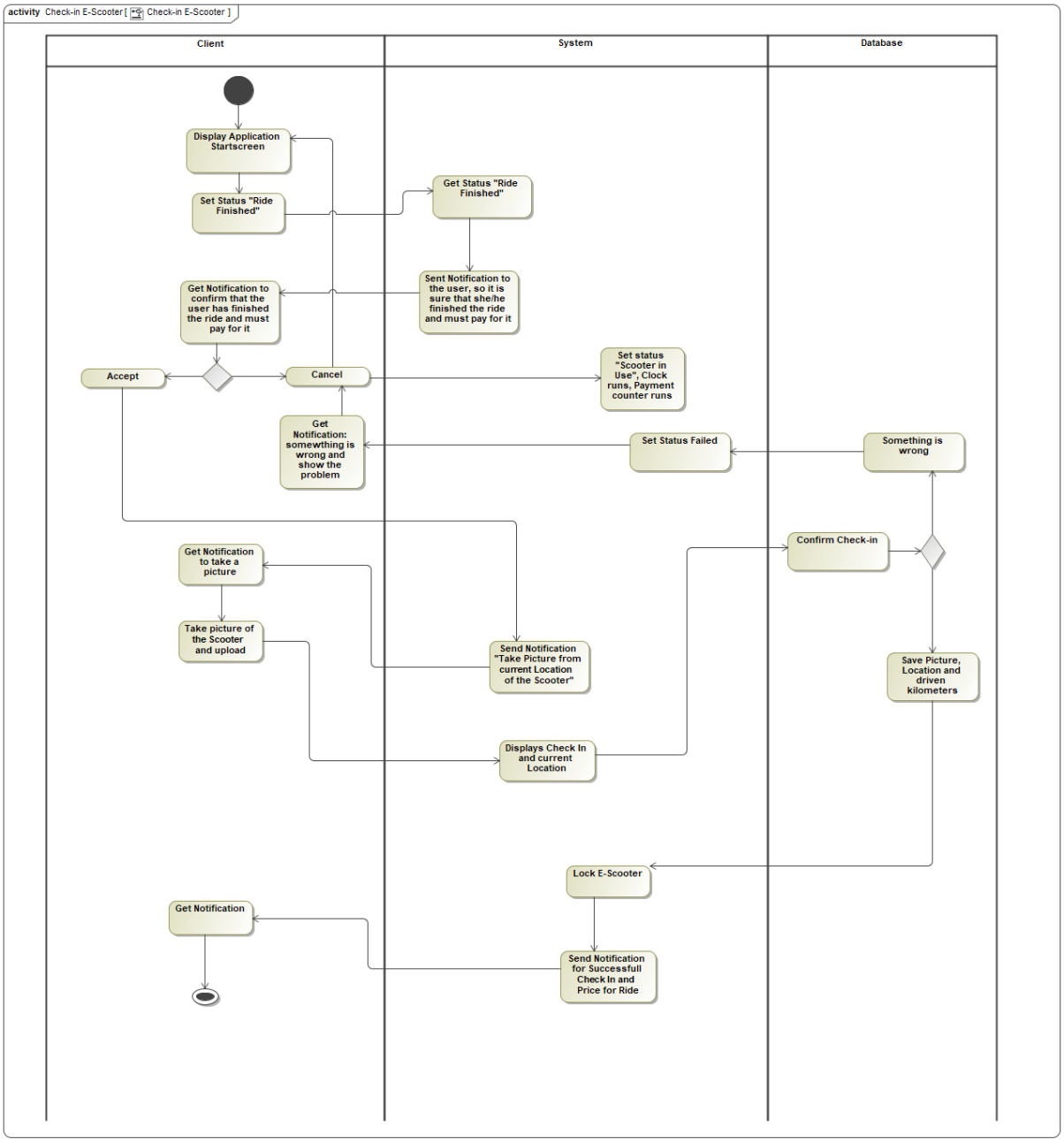


Figure 6. Check-in E-Scooter

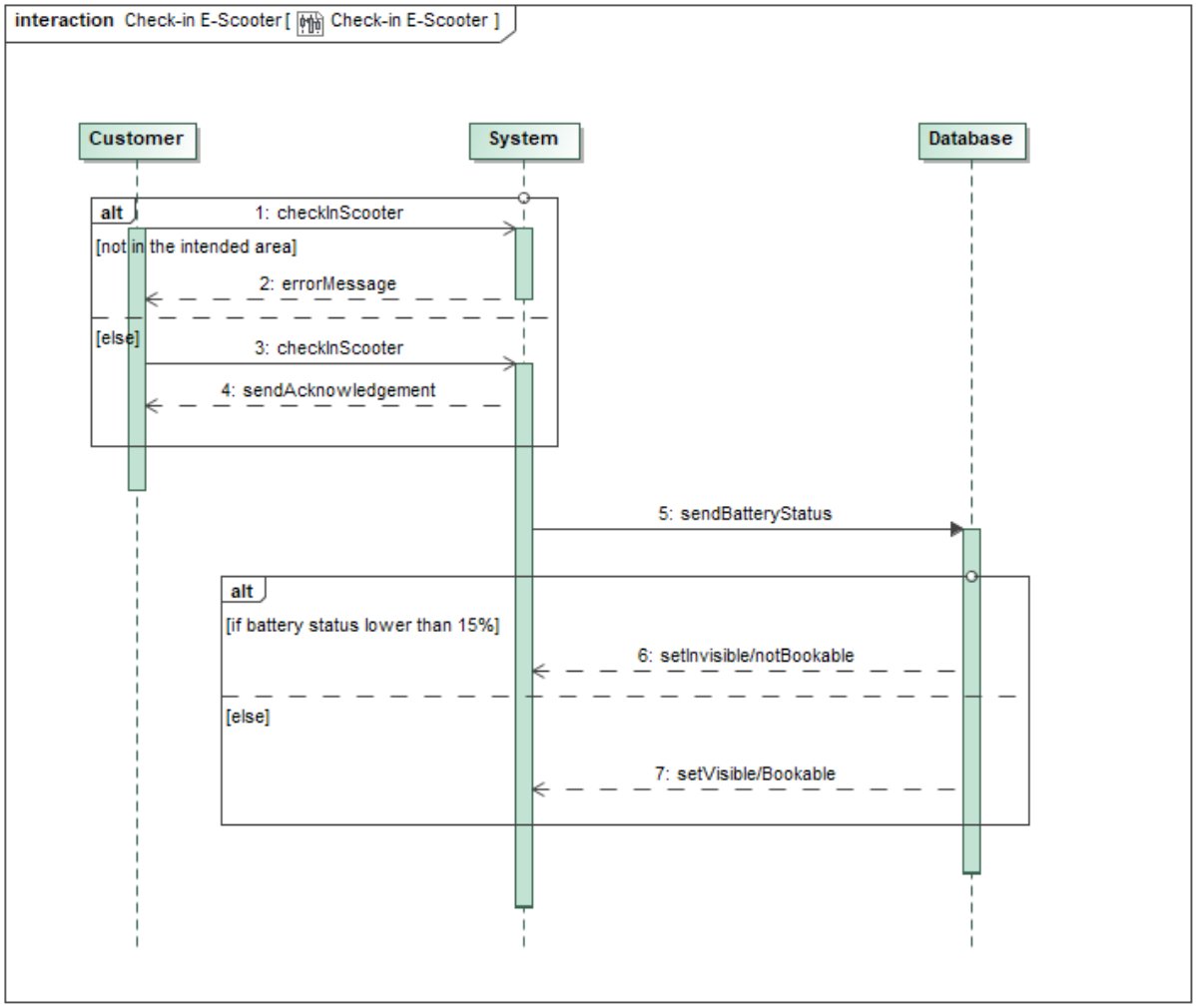


Figure 7. Check-in E-Scooter

10 UseCase Choose subscription plan

There are three membership tiers customers can subscribe: Gold, Silver and Bronze.

- Choose subscription plan includes Pay the service
- Choose subscription plan includes Pay via subscription plan
- Choose subscription plan is included by Subscribe to the service
- Choose subscription plan is included by Subscribe to the service

Extension Points

- Change Subscription Plan: Choose subscription plan

Description

Pre-Condition

Customer opened the app and is logged in.

Post-Condition

Customer changed the subscription plan successfully.

Goal

Give customers the option to choose a subscription plan which fits their needs.

Complexity

Average Complexity

Assumption

Customer did not choose a subscription yet.

Component Complexity

Low Complexity

Priority

High

Scenarios

Basic Flow

1. Customer opens the App and logs in.
2. Customer navigates through the main menu dropdown and finds the "Account Management".
3. Customer clicks on the current subscription status.
4. Customer chooses a plan according to her/his needs.

11 UseCase Change Subscription plan

Customers can change their membership type whenever they want. The new subscription then starts in the following month.

- Change Subscription plan extends Choose subscription plan

Description

Pre-Condition

Customer has logged in.

Post-Condition

Customer has successfully changed the subscription plan.

Goal

Give customers the option to adjust their subscription plan.

Complexity

Average Complexity

Assumption

Customer has already subscribed to the monthly E-Scooter service plan.

Component Complexity

Average Complexity

Priority

Normal

Scenarios

Basic Flow

1. Customer logs in.
2. Customer navigates to the "Account Management".
3. Customer changes her/his subscription plan.
4. UI refreshes.

12 UseCase Log-in

Registered customers must log-in to their account in the App in order to rent an E-Scooter.

- Log-in is performed by Customer
- Log-in is performed by Customer
- Log-in is performed by Customer
- Log-in includes Verify User Credentials

Extension Points

- Log-out: Log-in
- Login Errors: Log-in

Description

Pre-Condition

Customer has been successfully registered and has confirmed the Email address.

Post-Condition

Customer is logged in.

Goal

Customer logs in to ride an E-Scooter.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

1. Enter Email Address
2. Enter Password

3. Get Information
4. Validate Data

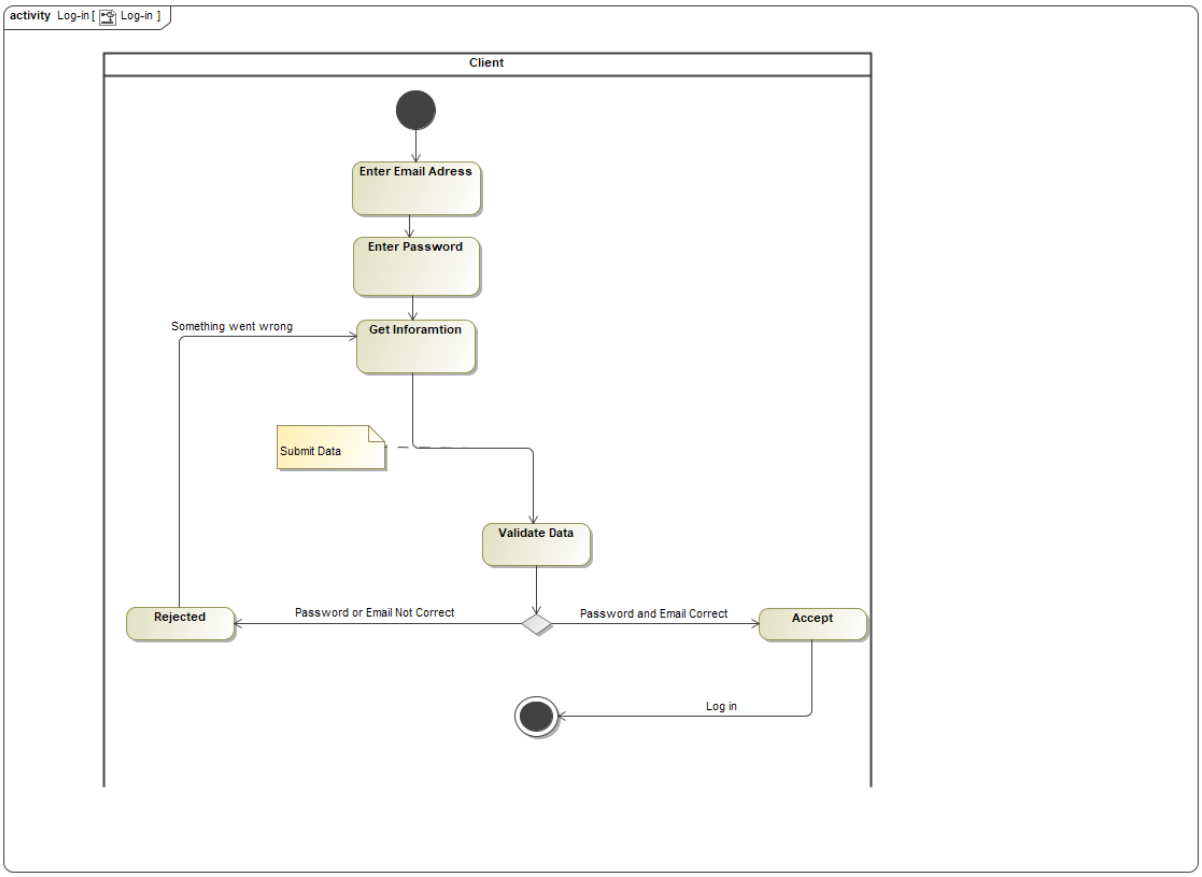


Figure 8. Log-in

13 UseCase Display Log-in Errors

When customers enter a wrong username or password, they get presented a log-in error.

- Display Log-in Errors extends Log-in

Description

Pre-Condition

Customer has been successfully registered.

Post-Condition

Customer is being presented with an alert warning her/him about the login errors.

Goal

When an incorrect password or username is entered the user should be presented with log in errors.

Complexity

Average Complexity

Component Complexity

Low Complexity

Priority

Above Normal

14 UseCase Verify User Credentials

The entered log-in data are compared and verified with those in the database.

- Verify User Credentials is included by Log-in

Description

Pre-Condition

Customer has successfully registered her/his account.

Post-Condition

Customer has entered the log-in credentials which have been verified.

Goal

Verify the log-in credentials entered by the customer.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

1. Customer enters the username and password.
2. Customer clicks log-in.
3. Log-in credentials are being compared with the log-in data in the database.

15 UseCase Log-out

When the customer has finished the ride and does not need the E-Scooter service any further, she/he can log-out of her/his account.

- Log-out is performed by Customer
- Log-out extends Log-in
- Log-out extends Log-in

Description

Pre-Condition

Customer is logged in.

Post-Condition

Customer is logged out.

Goal

Customer should be able to log out from the App.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

1. Open Menu
2. Tap "Logout" Button

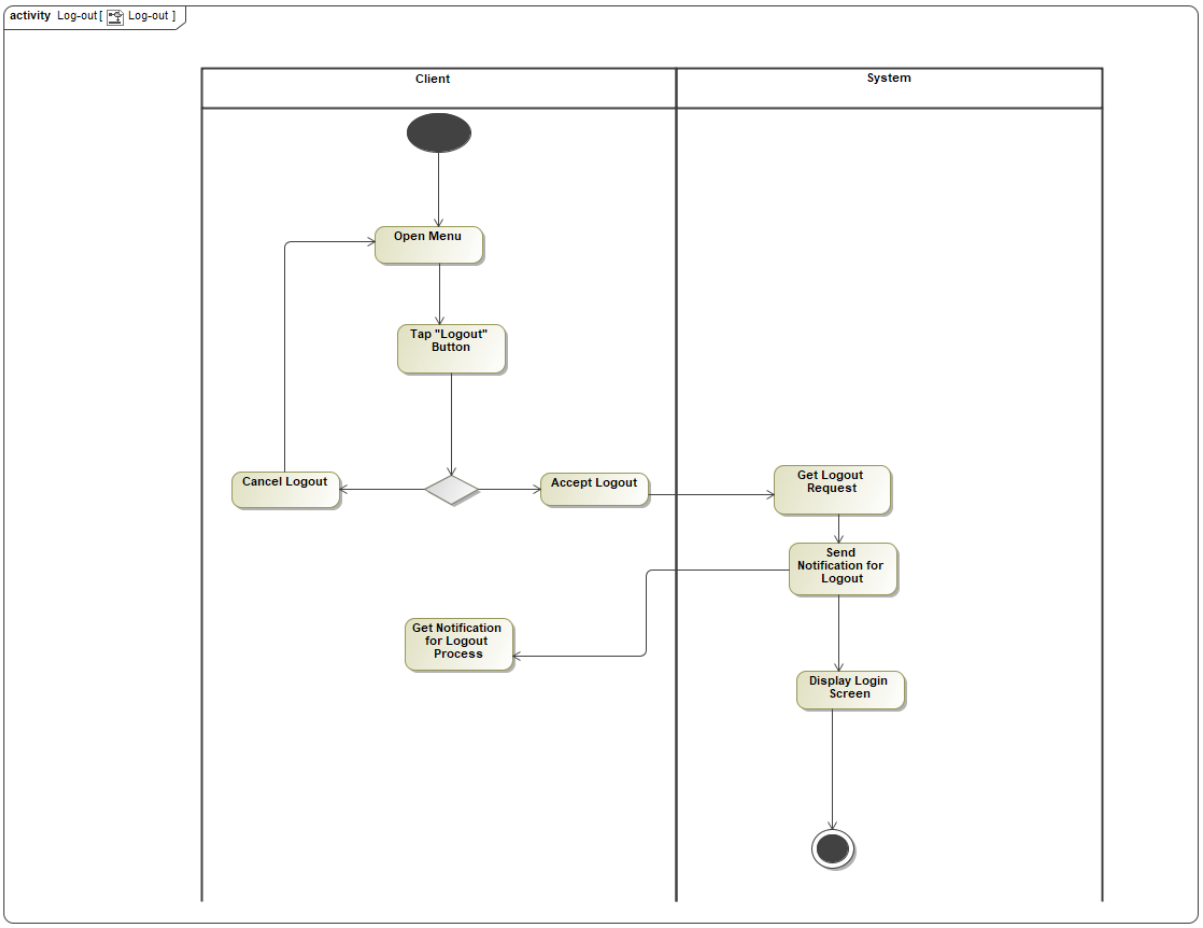


Figure 9. Log-out

16 UseCase Scan QR Code

Customers must scan the QR Code of an E-Scooter before the ride can start.

- Scan QR Code is included by Check-out E-Scooter

Description

Pre-Condition

Customer is logged in.

Post-Condition

QR-Code has been scanned.

Goal

Scan the Scooter's QR-Code in the Check-out process.

Complexity

Average Complexity

Assumption

Customer's phone has a camera.

Implementation Issue

No camera on the Customer's device.

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

1. Customer opens the App.
2. Customer gives permission to the App to use the camera.
3. Camera opens, and the QR-Code is being tracked by a red box.
4. QR-Code is scanned.
5. Availability attribute for the scanned Scooter is changed in the database.

17 UseCase Manage Account

Customers can change account details like bank credentials, E-Mail address, address, etc.

- Manage Account is performed by Customer
- Manage Account is performed by Customer

Extension Points

- Change Bank Credentials: Manage Account
- Check usage stats: Manage Account
- Delete Account: Manage Account

Description

Pre-Condition

Customer is logged in.

Post-Condition

Customer has saved changes to her/his account.

Goal

Customers can change their Personal details.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

1. Start Application
2. Open Account Management
3. Edit Account

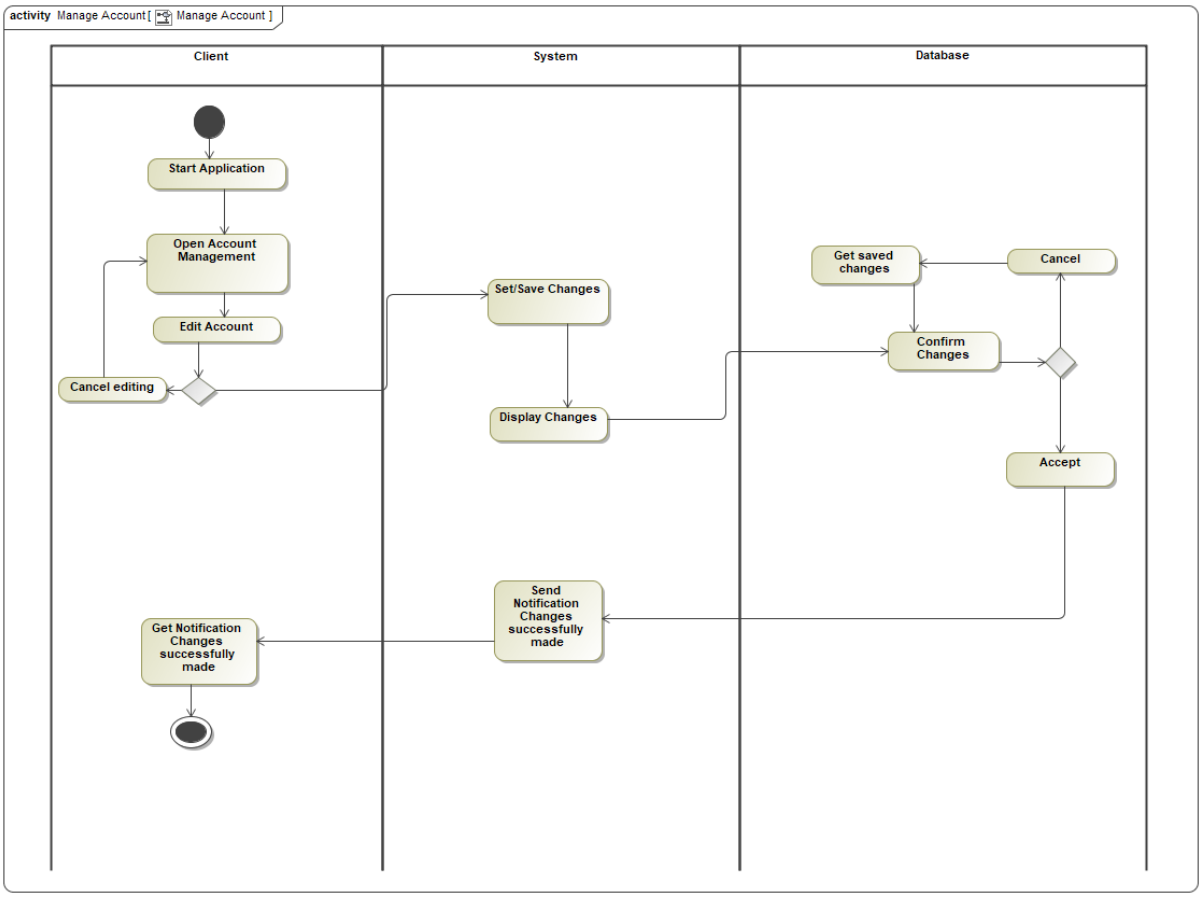


Figure 10. Manage Account

18 UseCase Delete Account

A customer can delete her/ his account, if she/ he does not want to use the service anymore.

Admins also can delete a customer's account, if she/ he violates the terms and conditions.

- Delete Account is performed by Customer
- Delete Account extends Manage Account

Description

Pre-Condition

Customer is logged in.

Post-Condition

Customer no longer possesses an account in the E-Scooter App.

Goal

Removes the customer's account from the database.

Complexity

Low Complexity

Assumption

Customer has created an account.

Component Complexity

Low Complexity

Priority

Above Normal

Scenarios

Basic Flow

1. Customer opens the App.
2. Customer navigates to the "Account Management".
3. Customer deletes the account.
4. Customer's information is removed from the App's database.

19 UseCase Find an E-Scooter on the map

Displays all nearby available E-Scooters on the map using a built-in map and GPS locator.

- Find an E-Scooter on the map is performed by Customer
- Find an E-Scooter on the map is performed by Customer
- Find an E-Scooter on the map is performed by Maintenance man
- Find an E-Scooter on the map includes Get GPS Location of the Customer
- Find an E-Scooter on the map includes Get GPS Location of the E-Scooter

Extension Points

- Download maps : Find an E-Scooter on the map
- Get GPS Location of the customer : Find an E-Scooter on the map
- Reserve an E-Scooter : Find an E-Scooter on the map

Description

Pre-Condition

App installed and Customer has made an account. GPS-Location has been allowed to share with the App.

Post-Condition

View of the map with nearby available Scooters on it.

Goal

Show the available scooters on the map for the customer's potential use.

Complexity

Average Complexity

Outstanding Issue

No available Scooters in the area.

Component Complexity

High Complexity

Priority

Above Normal

Scenarios

Basic Flow

1. Customer opens the App.
2. Customer gets presented with the map.

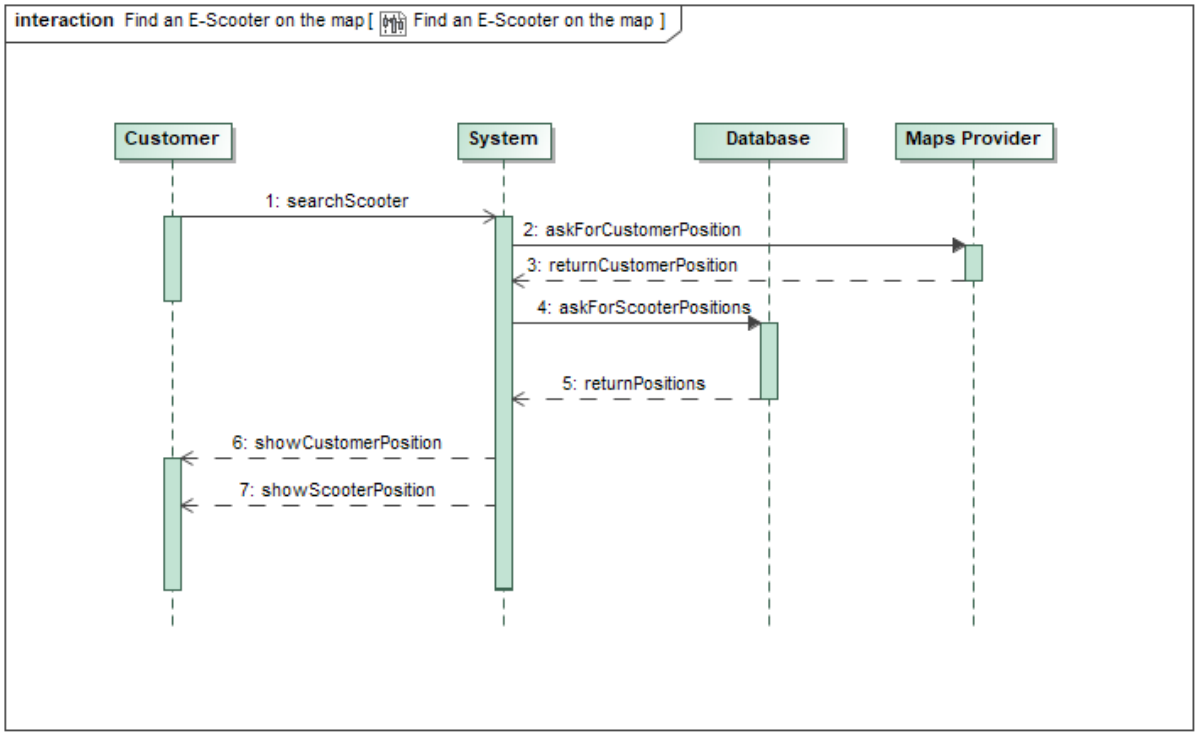


Figure 11. Find an E-Scooter on the map

20 UseCase Get GPS Location of the E-Scooter

Displays all nearby available E-Scooters on the map using a built-in map and GPS locator.

- Get GPS Location of the E-Scooter is performed by Administrator
- Get GPS Location of the E-Scooter is performed by Maintenance man
- Get GPS Location of the E-Scooter is performed by Maintenance man
- Get GPS Location of the E-Scooter is performed by Maps Service Provider
- Get GPS Location of the E-Scooter is included by Find an E-Scooter on the map

Description

Pre-Condition

Customer has opened the Application and turned on her/his GPS-Location service.

Post-Condition

Map updates and user gets the view of all nearby available Scooters.

Complexity

Average Complexity

Component Complexity

Low Complexity

Priority

Normal

Scenarios

Basic Flow

1. Customer opens the E-Scooter Service App.
2. Customer clicks "Allow sharing GPS-Location to this App".
3. Customer gets the view of all available Scooter nearby (relative to his/her position).

3.1 Maintenance man/Servicer gets presented with faulty or battery deficient Scooters.

Alternative Flow

1. User does not accept to share his GPS coordinates.
2. User gets presented with larger view of the City center that the entered as her/his city.

21 UseCase Get GPS Location of the Customer

Displays the customer's position on the map using GPS so she/ he is able to see their relative position to the nearby available E-Scooters.

- Get GPS Location of the Customer is performed by Maps Service Provider
- Get GPS Location of the Customer is included by Find an E-Scooter on the map
- Get GPS Location of the Customer extends Find an E-Scooter on the map

Description

Pre-Condition

Customer is logged in.

Post-Condition

Map updates showing the customer's position.

Goal

Map needs the current position of the customer in order to find E-Scooters on the map relative to her/his position.

Complexity

Average Complexity

Assumption

GPS Service is turned on the customer's end.

Component Complexity

Low Complexity

Priority

High

Scenarios

Basic Flow

1. Customer turns on the GPS Location service on the phone.
2. Map updates showing the customer's position.

Alternative Flow

1. GPS Location on the Customer's end is not turned on
2. Customer's in-app map gets updated showing the city center (taken from the Customer's account info)

22 UseCase Download Offline Maps

Customers can download an offline map of their area.

- Download Offline Maps is performed by Maps Service Provider
- Download Offline Maps is performed by Maps Service Provider
- Download Offline Maps extends Find an E-Scooter on the map
- Download Offline Maps extends Find an E-Scooter on the map

Description

Pre-Condition

Customer is logged in.

Post-Condition

Maps of the customer's surroundings in a n perimeter have been downloaded.

Goal

The customer should be able to download the maps from the Map Service Provides current surroundings in order to use it for an offline use.

Complexity

Average Complexity

Assumption

Customer has an Internet connection for the maps to be downloaded.

Component Complexity

Low Complexity

Priority

Low

23 UseCase Subscribe to the service

There are three membership tiers customers can subscribe: Gold, Silver and Bronze. Customers without a membership get their prices calculated per km or min/ h while the customers with a membership pay more affordable prices on a monthly/ yearly basis.

- Subscribe to the service is performed by Customer
- Subscribe to the service is performed by Customer
- Subscribe to the service includes Choose subscription plan
- Subscribe to the service includes Choose subscription plan
- Subscribe to the service extends Register

Extension Points

- Cancel Subscription: Subscribe to the service

Description

Pre-Condition

Customer is registered.

Customer is logged in.

Post-Condition

Customer chooses between many subscriptions plans available.

Goal

The option to subscribe and pay monthly instead after every ride should be allowed to the customer.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

Above Normal

24 UseCase Cancel Subscription

If a Customer knows she/ he will not use the service for a certain period of time, she/ he can cancel the membership without deleting the whole account.

- Cancel Subscription extends Subscribe to the service

Description

Pre-Condition

Customer has opened his/her app and is logged in.

Post-Condition

Customer's subscription record is removed from the database as the paying is switched to the wallet payment.

Goal

Delete a record in the database containing the customer's subscription details when the customer wants to cancel the subscription.

Complexity

Average Complexity

Component Complexity

Low Complexity

Priority

Above Normal

Scenarios

Basic Flow

1. Customer opens the app and is logged in.
2. Customer has navigated to the "Account Management/Change Account Details".
3. Customer has clicked his/her "Subscription button".
4. Customer clicks, confirms and cancels his/her subscription.

25 UseCase Pay the service

There are multiple ways for customers to pay for the service: Membership Subscription and Wallet.

- Pay the service is included by Check-in E-Scooter
- Pay the service is included by Choose subscription plan

Extension Points

- Subscription Plan Payment: Pay the service
- Wallet Payment: Pay the service

Description

Pre-Condition

Customer has checked-in the chosen E-Scooter.

Post-Condition

A payment transaction to the App's account has been made.

Goal

Make transactions for the given service between the Customer and the E-Scooter Service.

Complexity

Average Complexity

Assumption

Customer has at least n Euros on her/his in-app wallet.

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

1. Check-out an E-scooter
2. Check-in an E-Scooter
3. Confirm Check-in
4. Get Confirmation of Check-in
5. Display Drive Stats and Price
6. Request Info about Subscription
7. Retrieve Information about Subscription

Alternative Flow

7.1. Alternative Condition

7.1.1. Choose Payment method

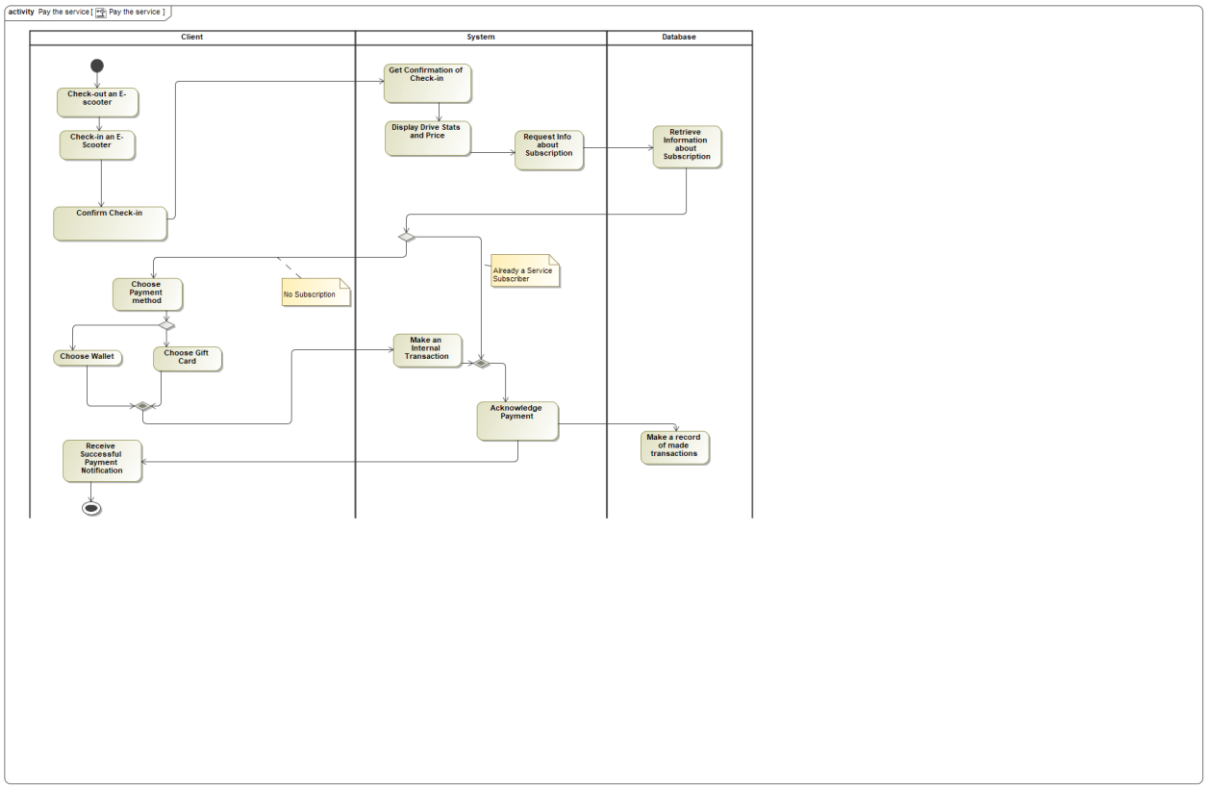


Figure 12. Pay the service

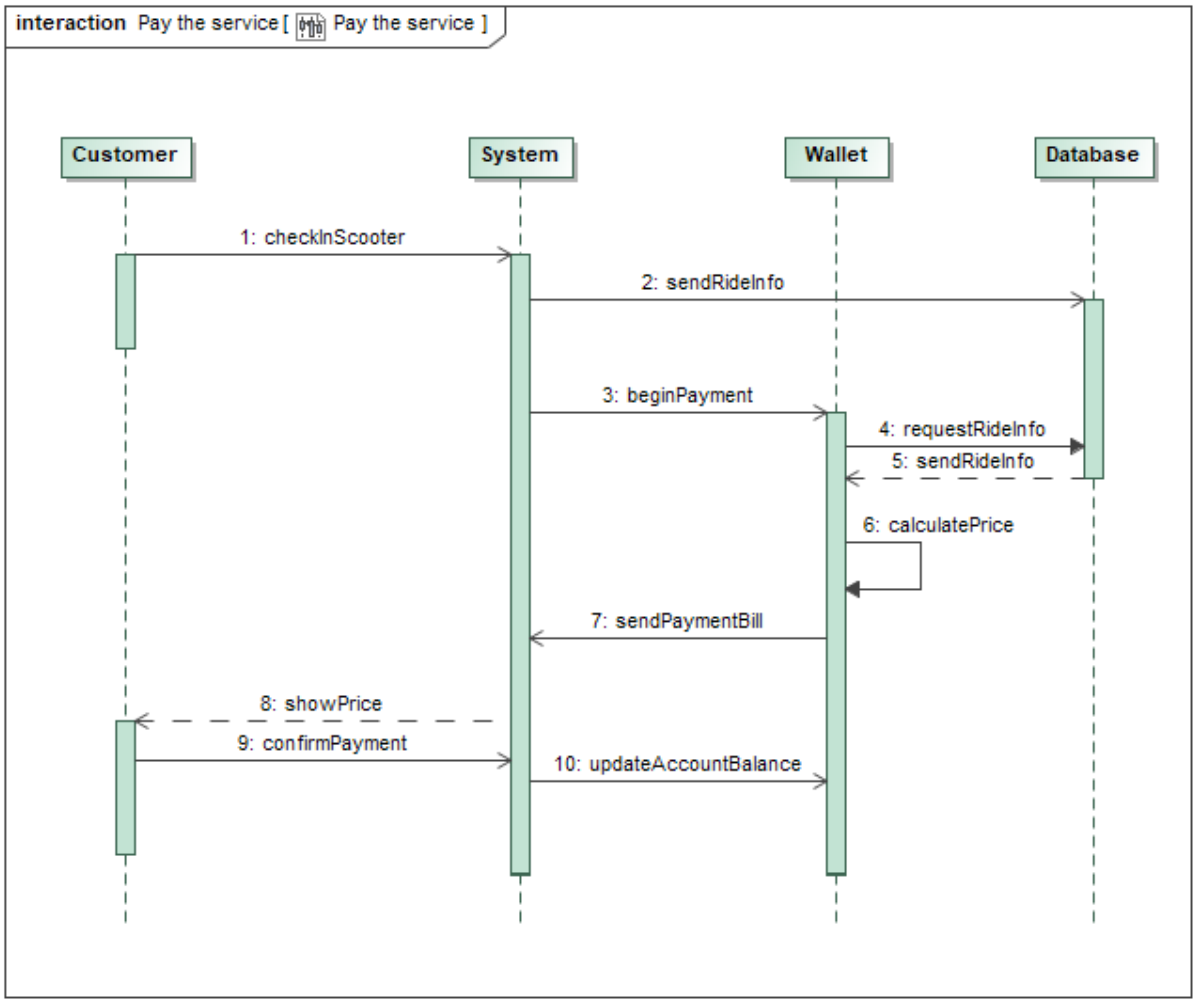


Figure 13. Pay the service

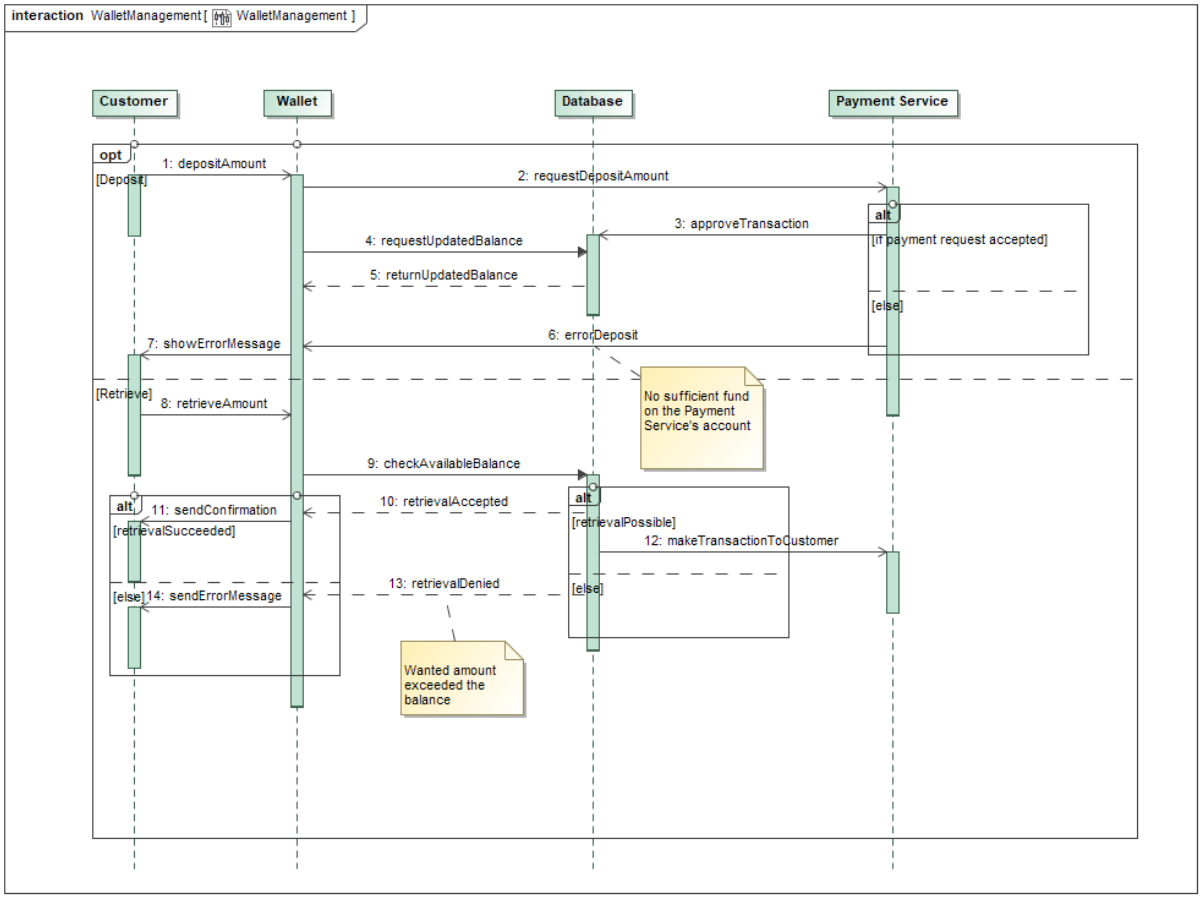


Figure 14. Wallet Management

26 UseCase Pay via subscription plan

Customers are able to choose to pay on a monthly or yearly basis by choosing among the three membership tiers: Gold, Silver and Bronze. The payment is made via a deposited credit/ debit card.

- Pay via subscription plan is performed by Payment Service
- Pay via subscription plan is included by Choose subscription plan
- Pay via subscription plan extends Pay the service

Description

Pre-Condition

Customer has checked-in her/his E-Scooter.

Post-Condition

Customer gets notified with the amount of km driven and the rest of her/his monthly subscription plan.

Goal

Customer has an option to pay a monthly fixed fee for a certain amount of km driven.

Complexity

Average Complexity

Assumption

Customer is subscribed to the E-Scooter service.

Customer has already entered the credit/debit card credentials.

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

1. Customer checks-in her/his E-Scooter.
2. System checks if the customer is a subscriber of the E-Scooter service.
3. Customer receives a notification of her/his ride details.

27 UseCase Pay via Wallet

The Wallet is an in-App virtual money storage system designed for the convenient use by the customer. It is needed for managing the customer's money balance for the ride.

- Pay via Wallet is performed by Payment Service
- Pay via Wallet extends Pay the service
- Pay via Wallet extends Pay the service

Description

Pre-Condition

Customer is logged in.

Customer has entered bank information.

Customer has deposited a minimum of n Euros to his/her in-App wallet.

Post-Condition

Customer's rides are being charged via Wallet.

Goal

Customer should be able to deposit and then later retrieve the money in order to make virtual wallet from which the rides will be charged.

Complexity

Average Complexity

Assumption

At least n Euros in the Wallet.

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

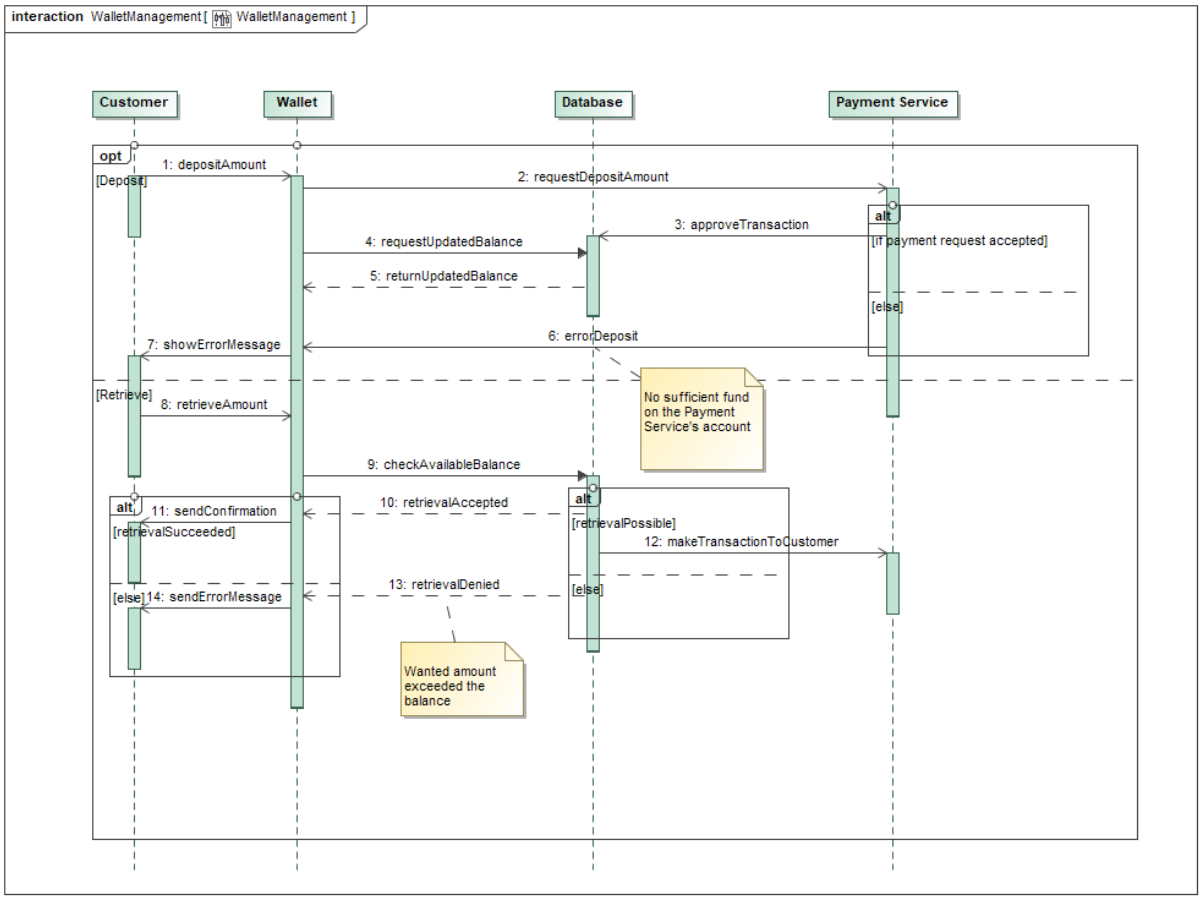


Figure 15. Basic flow of Pay via Wallet

28 UseCase Show E-Scooter Battery Status

Shows Admins/ Customers the battery status of each E-Scooter. On the hand, this helps the customers to decide, if the Scooter has enough battery for the ride, on the other hand, the Admins see the Scooters, which need to be recharged.

- Show E-Scooter Battery Status is performed by Administrator
- Show E-Scooter Battery Status is performed by Maintenance man
- Show E-Scooter Battery Status is included by Check-out E-Scooter

Description

Goal

Show battery status of the E-Scooters on the map to any user type.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

High

29 UseCase Reserve an E-Scooter

Customers have the possibility to pre-book an E-Scooter for later or for another day.

- Reserve an E-Scooter is performed by Customer
- Reserve an E-Scooter extends Find an E-Scooter on the map

Description

Pre-Condition

Customer is logged in.

Customer has found an E-Scooter that he/she wants to reserve.

Post-Condition

A chosen E-Scooter has been reserved.

Goal

Customer should be able to make a reservation for an E-Scooter found on the map.

Complexity

Average Complexity

Assumption

Scooter to be reserved is available.

Component Complexity

Average Complexity

Priority

Normal

Scenarios

Basic Flow

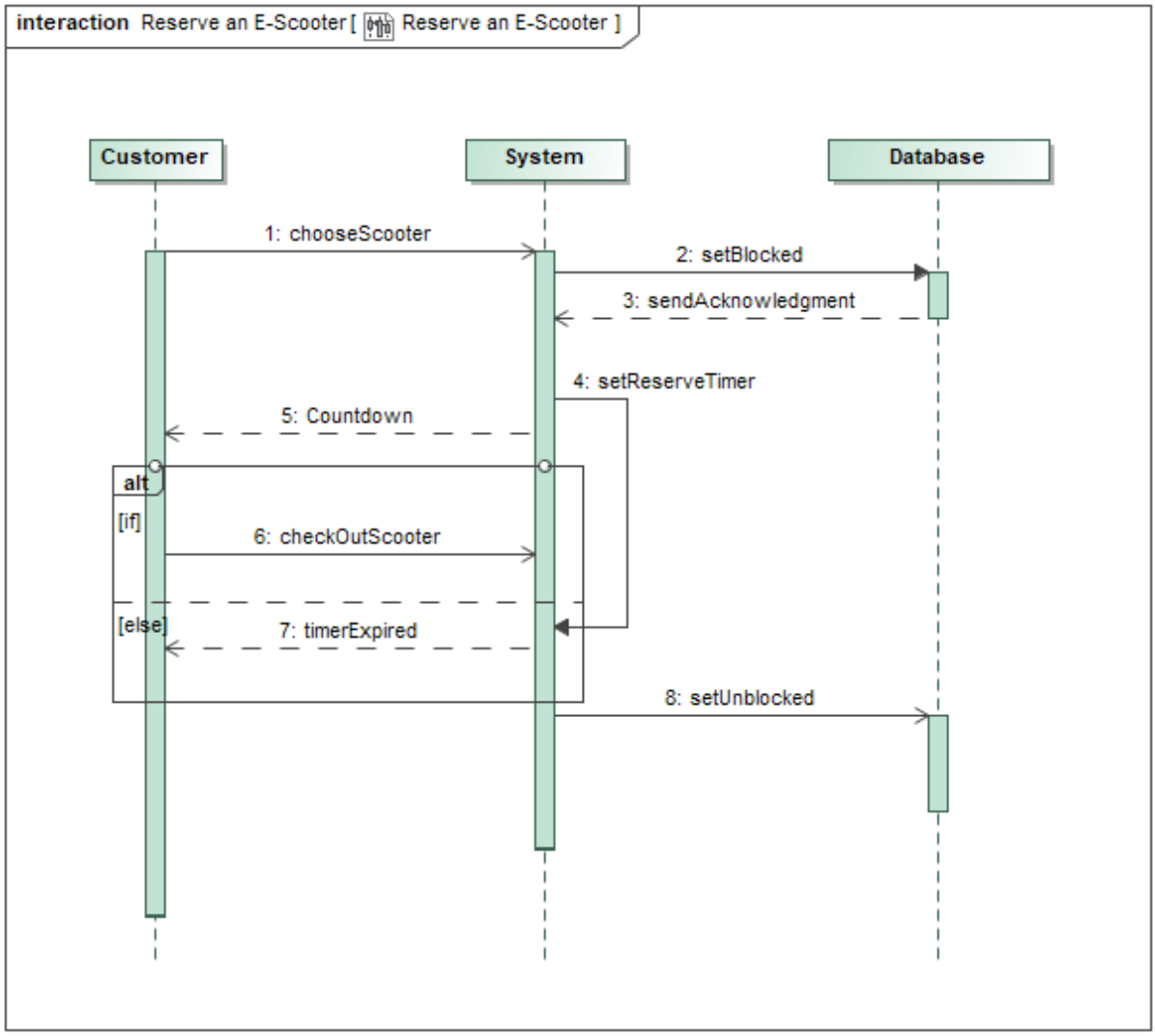


Figure 16. Reserve an E-Scooter

30 UseCase Report Problems with Scooter

Customers can send information about specific problems with an E-Scooter.

- Report Problems with Scooter is performed by Customer
- Report Problems with Scooter is performed by Customer

Description

Pre-Condition

Customer/Maintenance man is logged in.

Post-Condition

A report has been filled detailing a problem with(-out) pictures attached to it.

Goal

Make reports about the potential and current problems with an E-Scooter.

Complexity

Average Complexity

Assumption

There is a problem with an E-Scooter

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

1. Open the App and find "Report a Problem" option in the Menu
2. File A Basic Report and/or attach a Photo
3. Get the Problem Report
4. Store the Reports
5. Analyze Reports

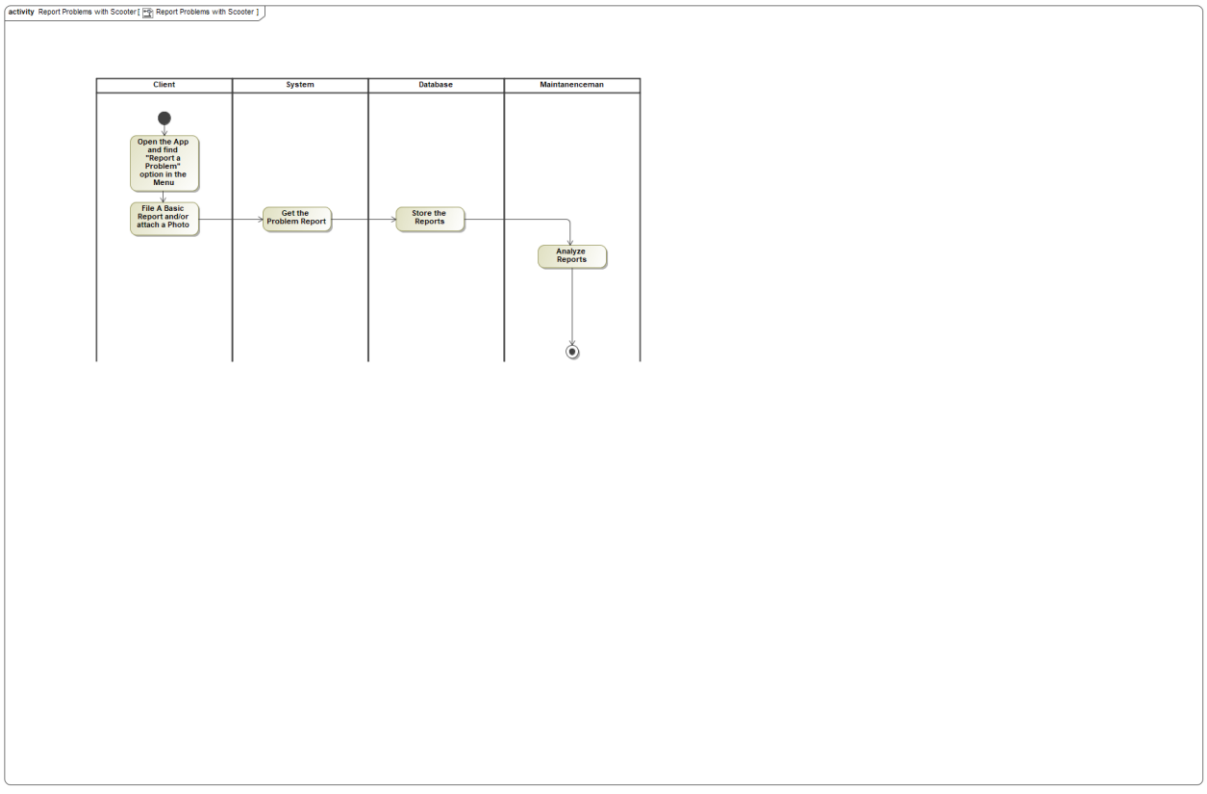


Figure 17. Report Problems with Scooter

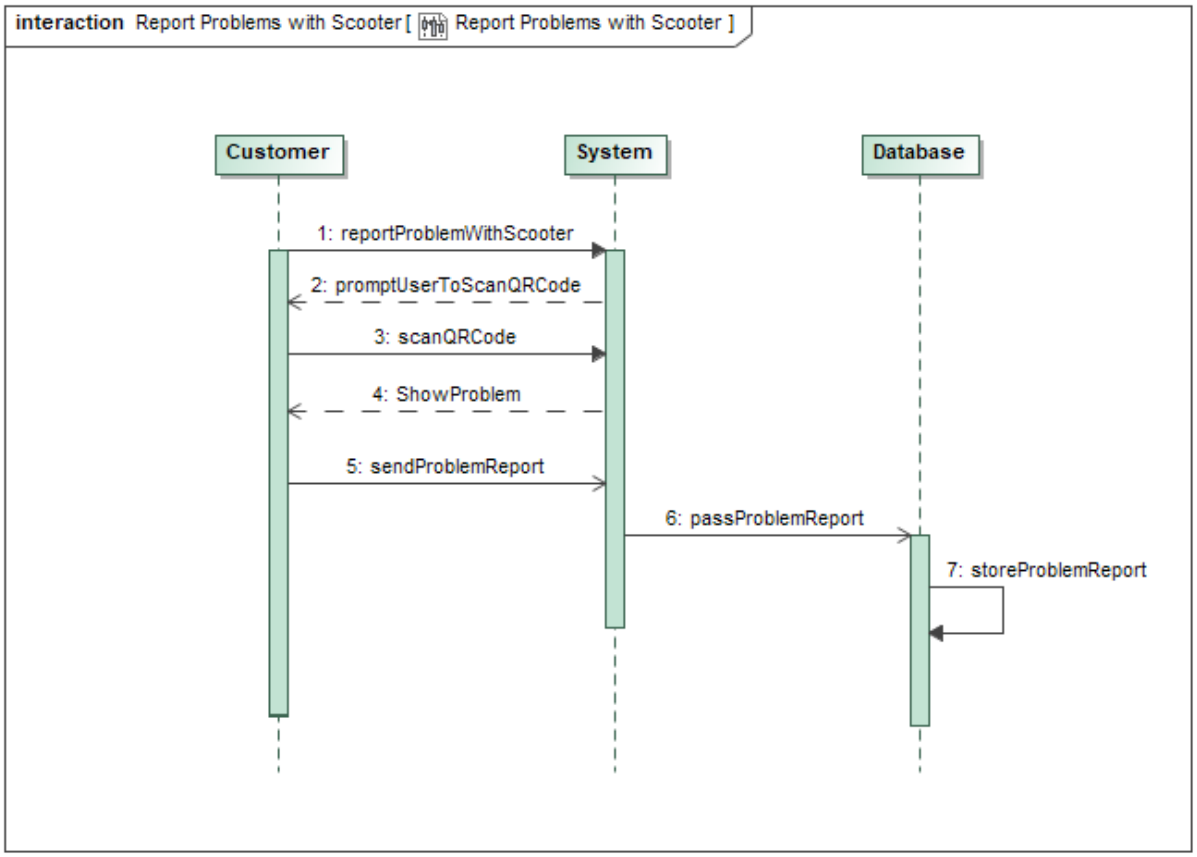


Figure 18. Report Problems with Scooter

31 UseCase Give Feedback

If customers have any suggestions to improve the service, they can give a feedback in the App.

- Give Feedback is performed by Customer
- Give Feedback is performed by Customer
- Give Feedback includes Analyze Customer Feedback

Description

Pre-Condition

Customer is logged in.

Post-Condition

Customer has left a feedback on the current situation of the app in form of a comment.

Goal

Customer can leave a feedback in the app at any time.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

High

Scenarios

Basic Flow

1. Leave a Feedback on the App
2. Store Customer Feedback
3. Retrieve Stored Feedbacks
4. Analyze Customer Feedback

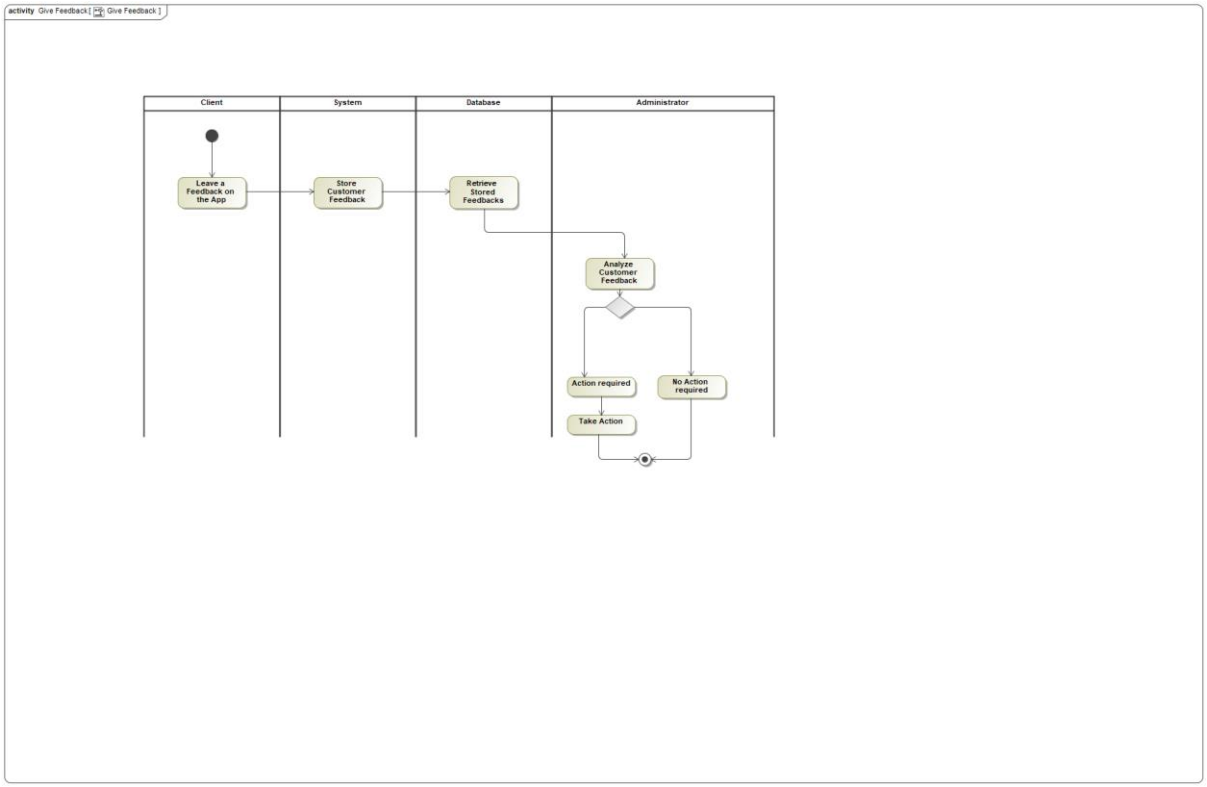


Figure 19. Give Feedback

32 UseCase Ask for feedback

- Ask for feedback is performed by Administrator
- Ask for feedback extends Get Notifications

Description

Pre-Condition

Customer has opened the app.

Post-Condition

Customer gets presented with a panel containing the option to leave a feedback on the app/service.

Goal

Get feedback through a panel in the app UI containing a star scale and the possibility to add a comment for the Customer.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

Below Normal

33 UseCase Analyze Customer Feedback

By analyzing customer feedbacks, the company can improve the service and keep the customer satisfaction on a high level.

- Analyze Customer Feedback is performed by Administrator
- Analyze Customer Feedback is included by Give Feedback

Description

Pre-Condition

Admin has received feedbacks

Post-Condition

According to feedbacks with comments admin computes the result of the most wanted customer changes/adjustments to the service/app.

Goal

Analyze the current situation of the app.

Complexity

Low Complexity

Component Complexity

Average Complexity

Priority

Normal

34 UseCase Send warning

There is a well-defined "legal" area, where the E-Scooters can be driven and checked-in.

If the customer is leaving this "legal" area, she/ he is getting a push notification on her/ his device with a warning, that she/ he is leaving the allowed area and that the Scooter will be locked, if she/ he does not drive back to the "legal" area.

- Send warning extends Get Notifications

Description

Pre-Condition

Customer has checked-out her/his Scooter

Customer drove out of bounds.

Post-Condition

Customer gets a warning notification telling him/her that the Scooter will be locked if the Customer does not drive back into allowed limits.

Goal

Send a warning message telling the user that he/she drove out of the allowed limits or that the user is driving inappropriately.

Complexity

Average Complexity

Assumption

Driving area limit is well defined and possibly presented on the map.

Component Complexity

Average Complexity

Priority

Above Normal

35 UseCase Get Notifications

Customers get a push notification on their devices via the App.

- Get Notifications is performed by Customer

Extension Points

- Push notifications: Get Notifications
- Warnings: Get Notifications

Description

Pre-Condition

Customer is registered.

Post-Condition

Customer receives notifications from the App.

Goal

Customer receives push notifications/newsletter/warnings from the App.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

Below Normal

36 UseCase Send a push notification

Send push notifications to the customer's device via the App.

Customers get notifications about:

- Driving rules: all new customers are shown the basic driving rules before their first ride with an E-Scooter.

- Safety risks: customers are showed the safety risks of driving an E-Scooter.

- Legal information: if a customer tries to misuse the E-Scooter, she/ he gets presented with the legal constraints and risks.

- Newsletter: Customers receive an E-Mail with news about the App, special offers, etc.

- Send a push notification is performed by Administrator
- Send a push notification extends Get Notifications
- Send a push notification extends Get Notifications

Description

Goal

Customer should be able to receive push notifications.

Complexity

Average Complexity

Component Complexity

Average Complexity

Priority

Normal

37 UseCase Send an email confirmation

After a customer has entered all relevant registration information, she/ he gets an Email confirmation to complete the registration.

- Send an email confirmation is included by Register

Description

Pre-Condition

Customer has entered all relevant registration details.

Post-Condition

An Email confirmation of the registration has been sent to the customer's Email address.

Goal

Customer's Email address should be confirmed upon registration.

Complexity

Average Complexity

Assumption

Customer has entered a right Email address belonging to him/her.

Component Complexity

Average Complexity

Priority

Below Normal

E-scooter App

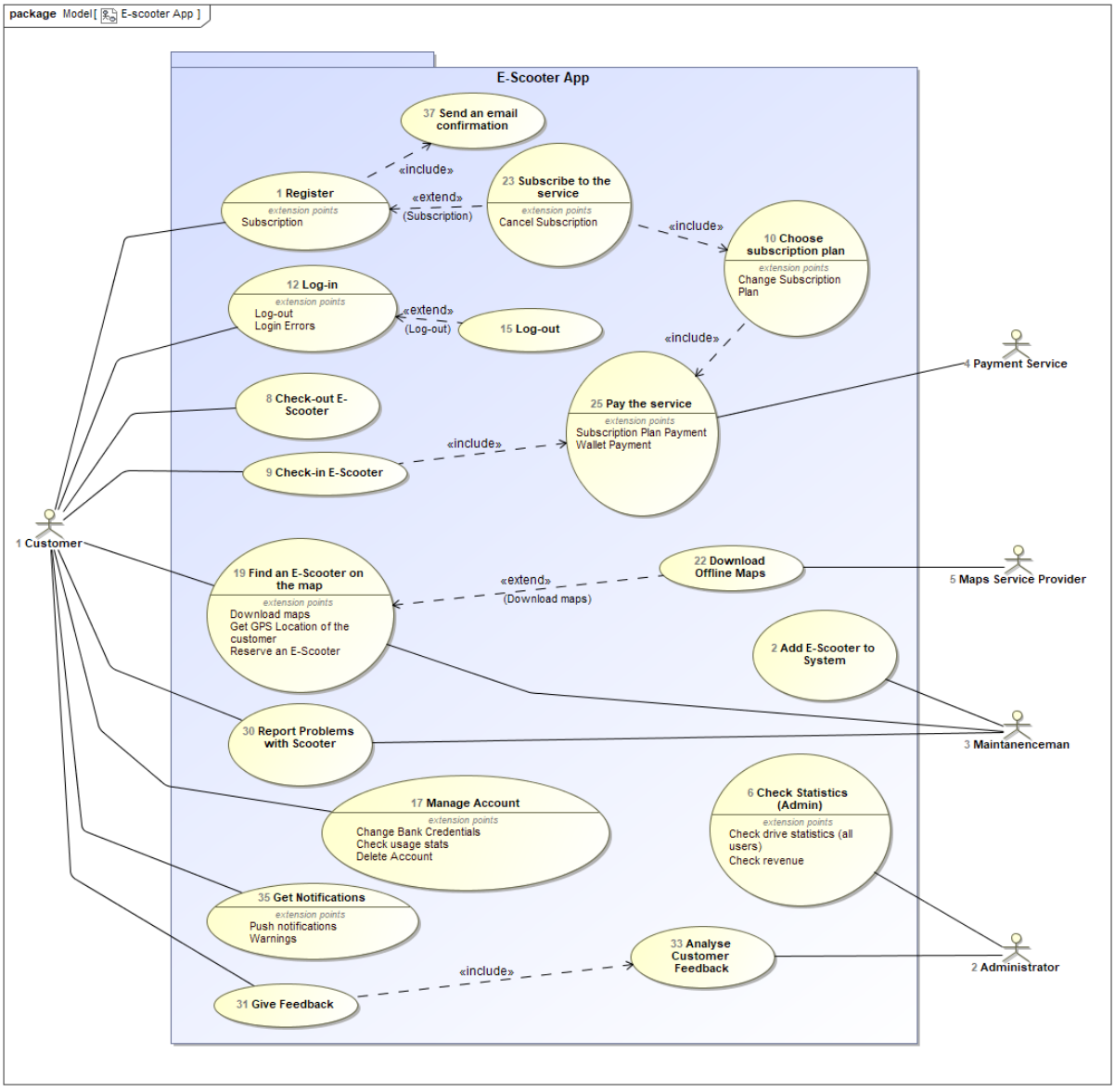


Figure 20. E-scooter App

Actor Administrator

- Administrator performs Analyze Customer Feedback
- Administrator performs Ask for feedback
- Administrator performs Check Statistics (Admin)
- Administrator performs Check Statistics (Admin)
- Administrator performs Get GPS Location of the E-Scooter
- Administrator performs Send a push notification
- Administrator performs Show E-Scooter Battery Status

Actor Customer

- Customer performs Check usage statistics (Customer)
- Customer performs Check usage statistics (Customer)
- Customer performs Check-in E-Scooter
- Customer performs Check-out E-Scooter
- Customer performs Check-out E-Scooter
- Customer performs Delete Account
- Customer performs Delete Account
- Customer performs Find an E-Scooter on the map
- Customer performs Find an E-Scooter on the map
- Customer performs Get Notifications
- Customer performs Get Notifications
- Customer performs Give Feedback
- Customer performs Give Feedback
- Customer performs Log-in
- Customer performs Log-in
- Customer performs Log-in
- Customer performs Log-out
- Customer performs Manage Account
- Customer performs Manage Account
- Customer performs Pay the service
- Customer performs Register
- Customer performs Report Problems with Scooter
- Customer performs Report Problems with Scooter

- Customer performs Reserve an E-Scooter
- Customer performs Subscribe to the service
- Customer performs Subscribe to the service

Actor Maintenance man

- Maintenance man performs Add E-Scooter to System
- Maintenance man performs Find an E-Scooter on the map
- Maintenance man performs Get GPS Location of the E-Scooter
- Maintenance man performs Get GPS Location of the E-Scooter
- Maintenance man performs Report Problems with Scooter
- Maintenance man performs Show E-Scooter Battery Status
- Maintenance man performs Show E-Scooter Battery Status

Actor Maps Service Provider

- Maps Service Provider performs Download Offline Maps
- Maps Service Provider performs Download Offline Maps
- Maps Service Provider performs Get GPS Location of the Customer
- Maps Service Provider performs Get GPS Location of the E-Scooter

Actor Payment Service

- Payment Service performs Pay the service
- Payment Service performs Pay via Wallet
- Payment Service performs Pay via Wallet
- Payment Service performs Pay via subscription plan

Known other usecases

UseCase Register, UseCase Add E-Scooter to System, UseCase Check Statistics (Admin), UseCase Check-out E-Scooter, UseCase Check-in E-Scooter, UseCase Choose subscription plan, UseCase Log-in, UseCase Log-out, UseCase Manage Account, UseCase Find an E-Scooter on the map, UseCase Download Offline Maps, UseCase Subscribe to the service, UseCase Pay the service, UseCase Report Problems with Scooter, UseCase Give Feedback, UseCase Analyze Customer Feedback, UseCase Get Notifications, UseCase Send an email confirmation

11.2 Detailed Use Case Diagrams

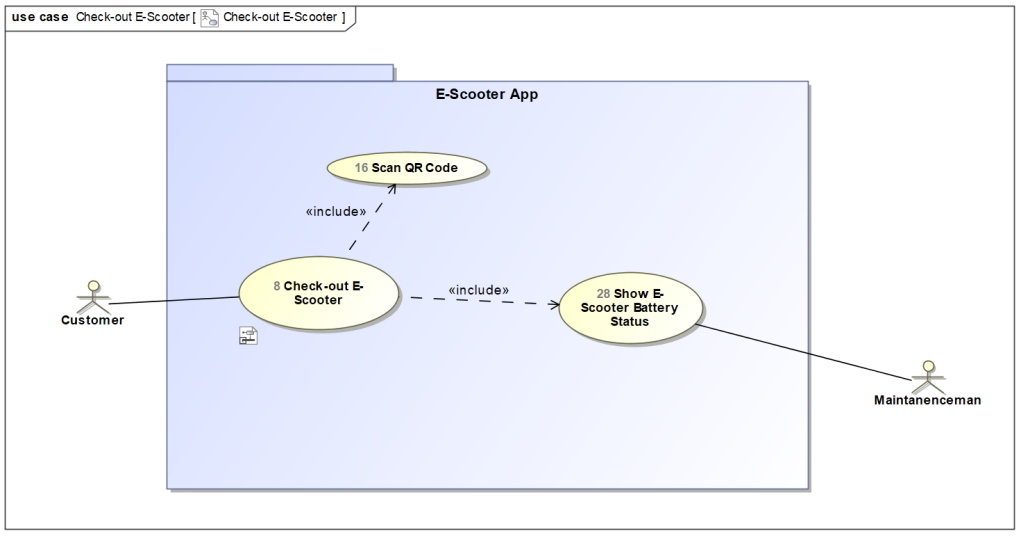


Figure 2: Use Case: Check-Out E-Scooter

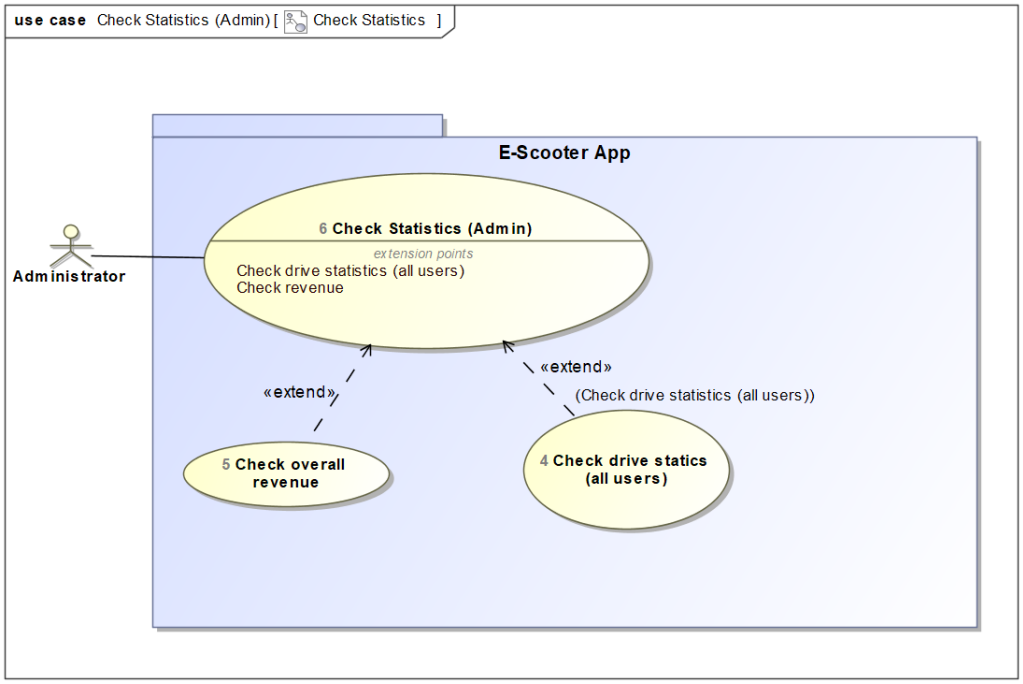


Figure 3: Use Case: Check Statistics (Admin)

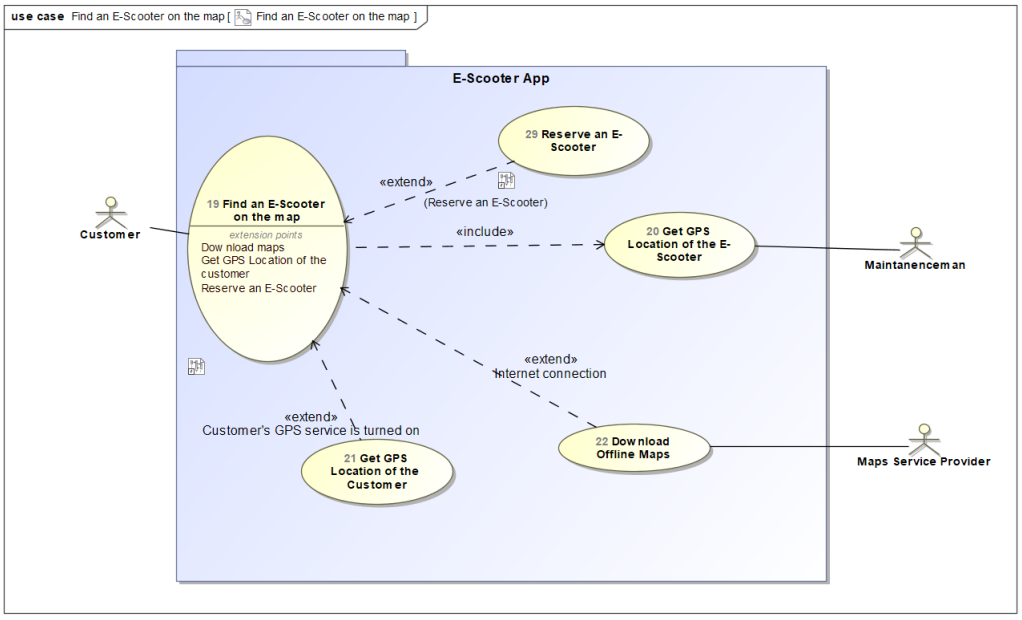


Figure 4: Use Case: Find an E-Scooter on the map

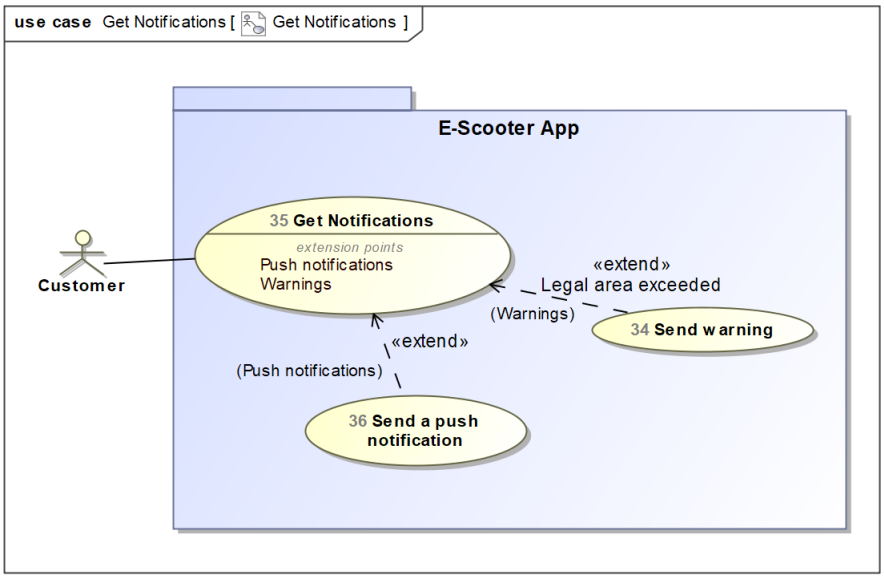


Figure 5: Use Case: Get Notifications

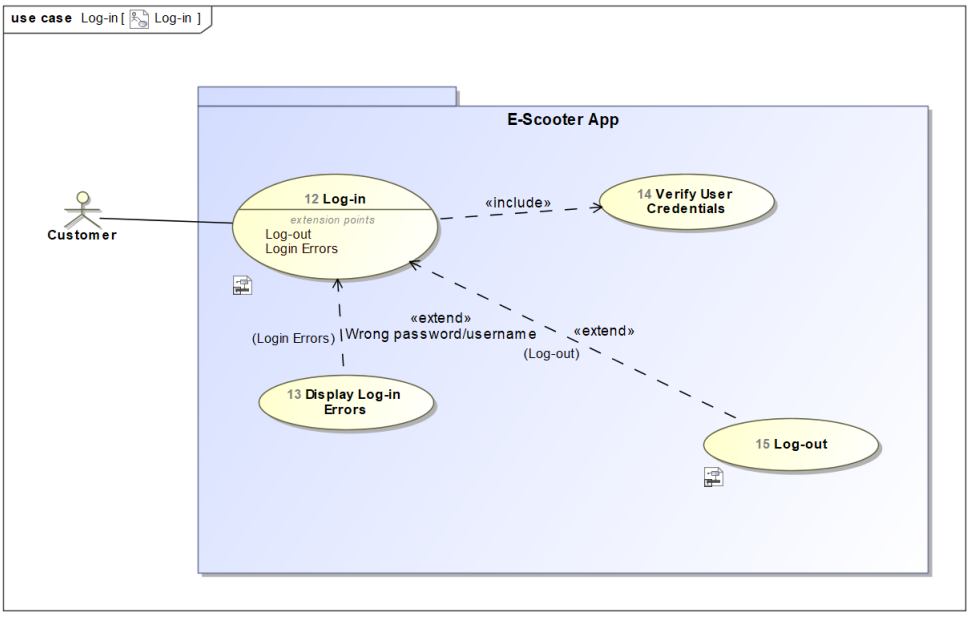


Figure 6: Use Case: Log-in

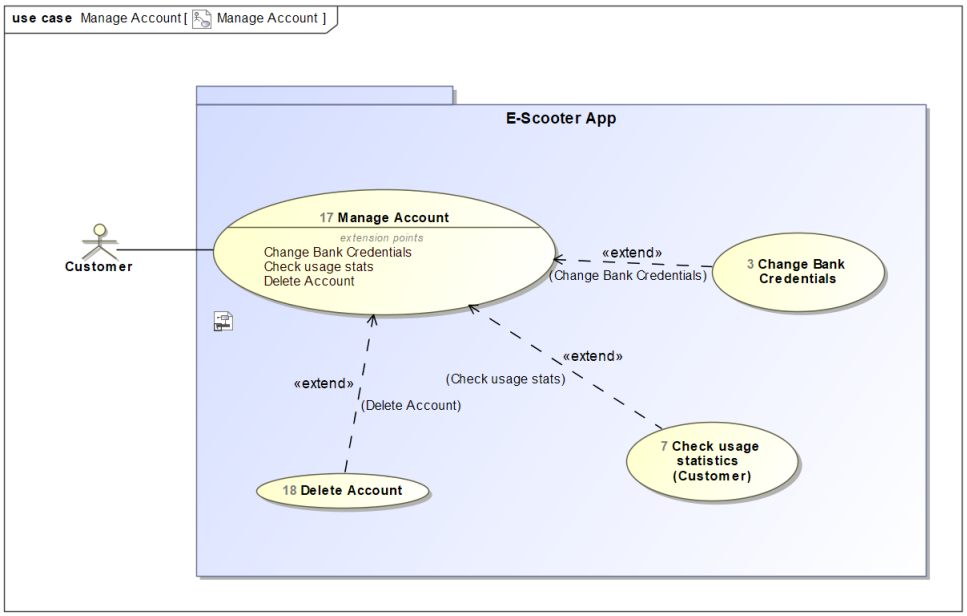


Figure 7: Use Case: Manage Account

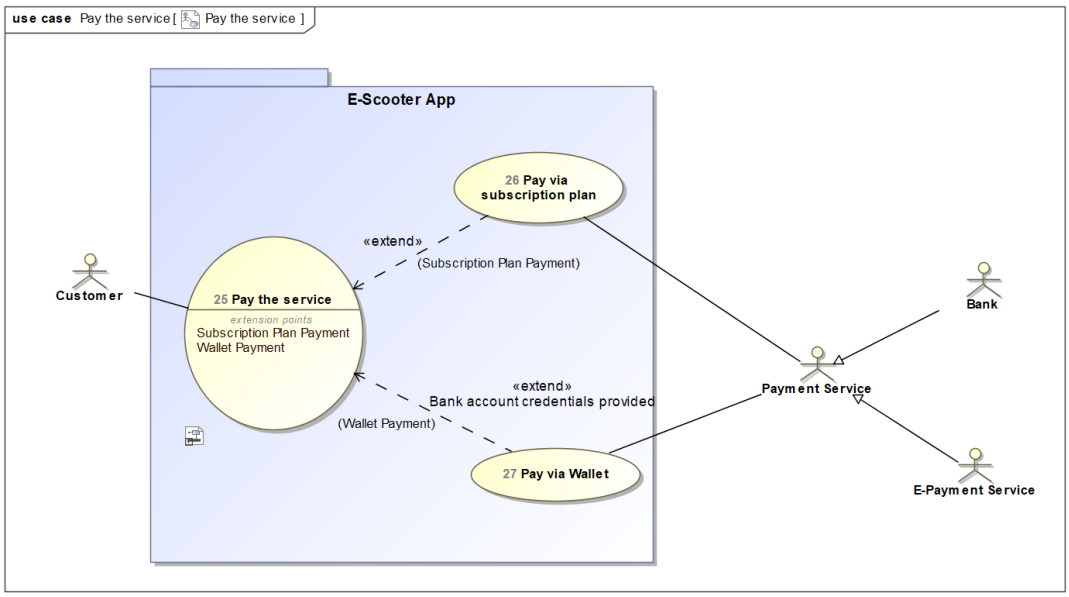


Figure 8: Use Case: Pay The Service

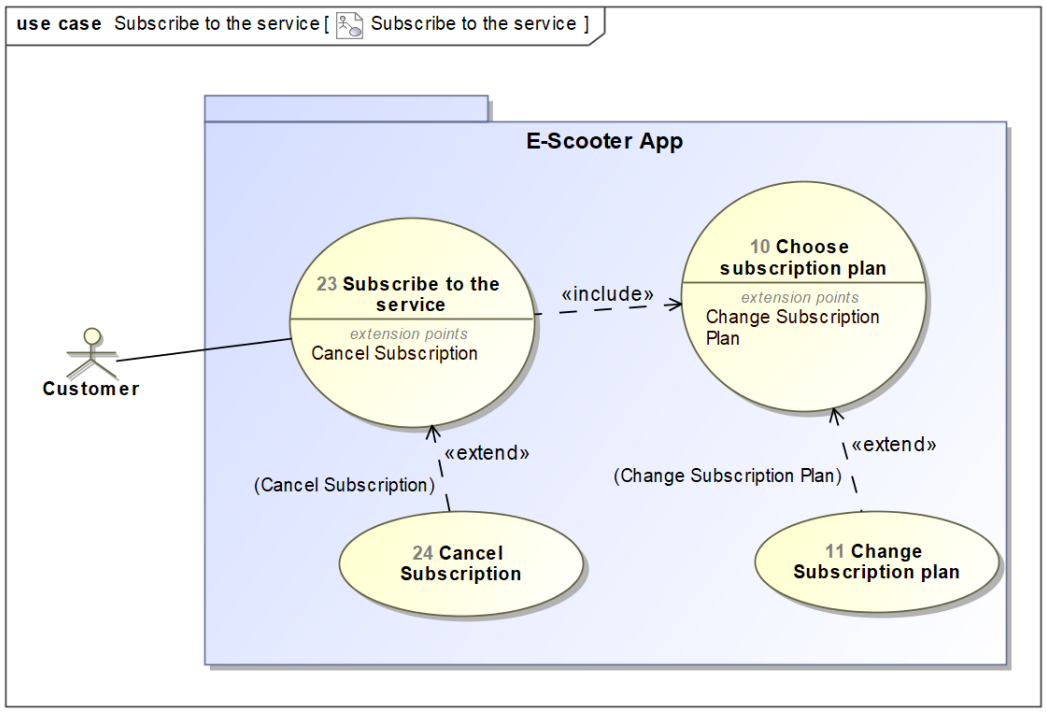
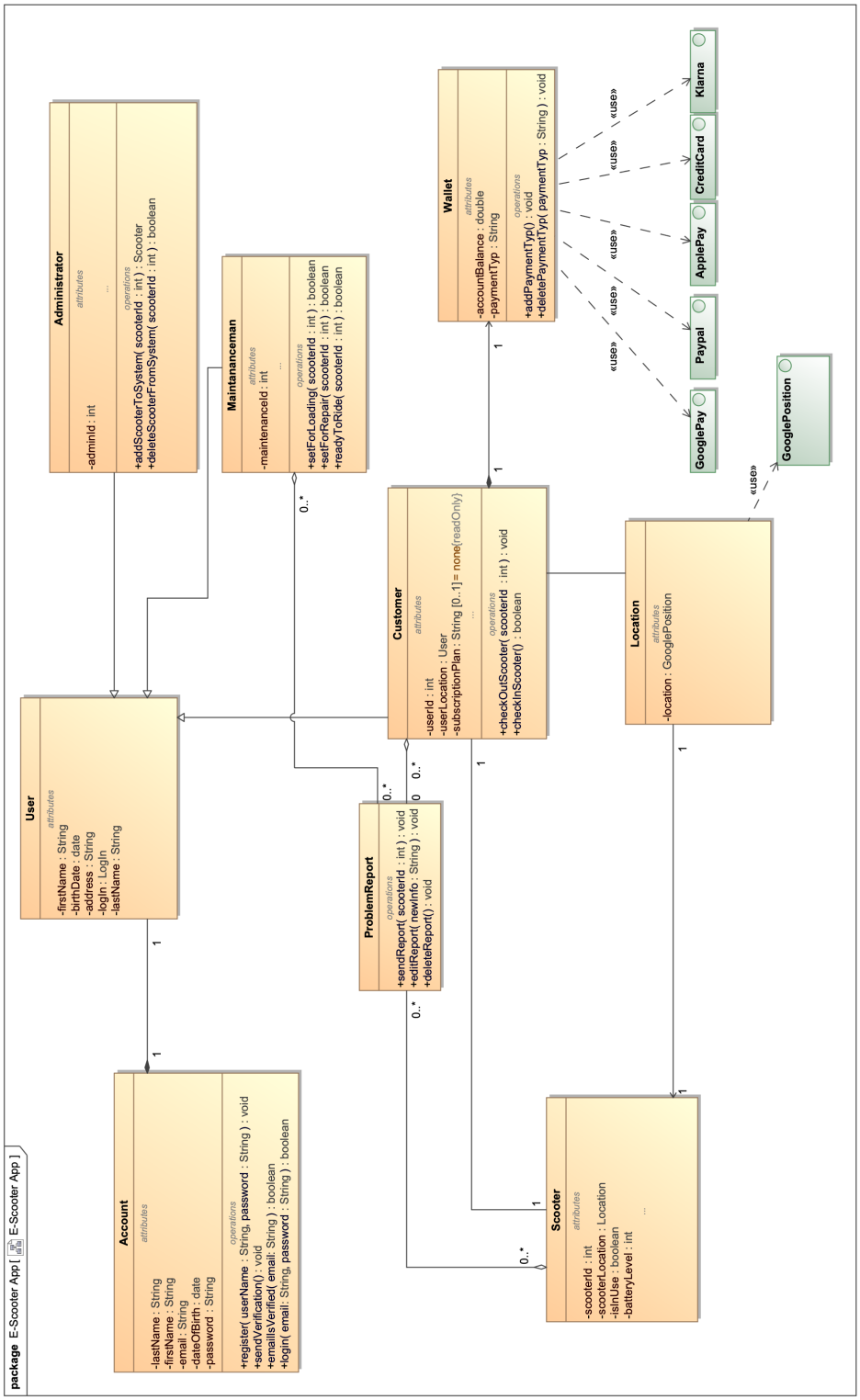


Figure 9: Use Case: Subscribe to The Service

11.3 Class Diagram

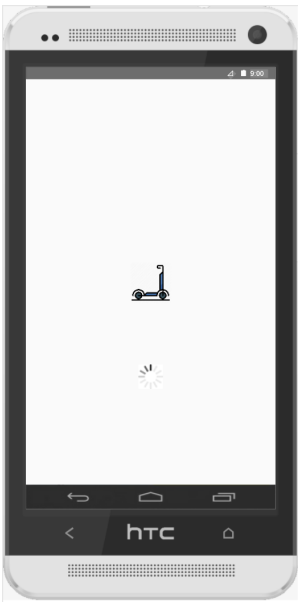


11.4 Presentation Slides

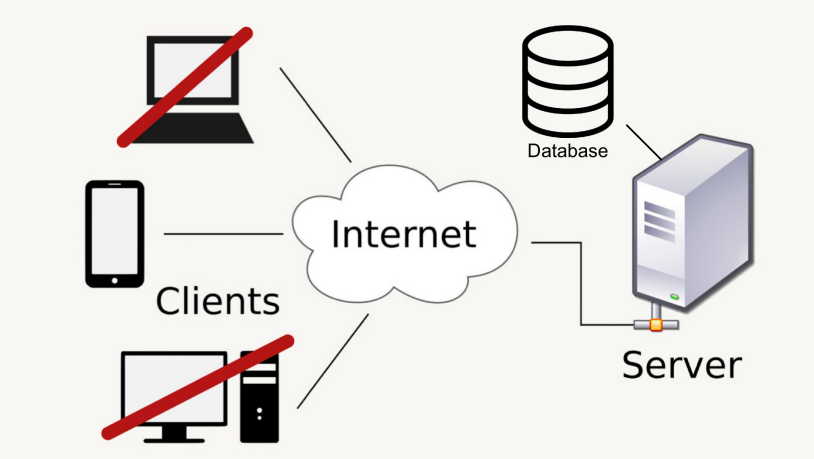
Software Engineering – Analysis Project: E-Scooter Rental Service

- Kendra Birringer (1229372)
- Nader Cacace (1208115)
- Steffen Hanzlik (1207417)
- Marco Peluso (1228849)
- Svetozar Stojanovic (1262287)

Frankfurt University of Applied Sciences



High-level Software Architecture



Source: https://commons.wikimedia.org/wiki/File:Client-server-model.svg
(modified by Kendra Birringer)

Tools

- Google Sheets
- Discord
- GitHub
- MagicDraw
- Axure
- LaTeX

Google Sheets

Source: https://de.cleanpng.com/png-8ht5bw/

Source: https://discordapp.com/branding

Source: https://github.com/logos

Source: https://www.lieberlieber.com/embedded-engineer/home-2/magicdraw_logo/

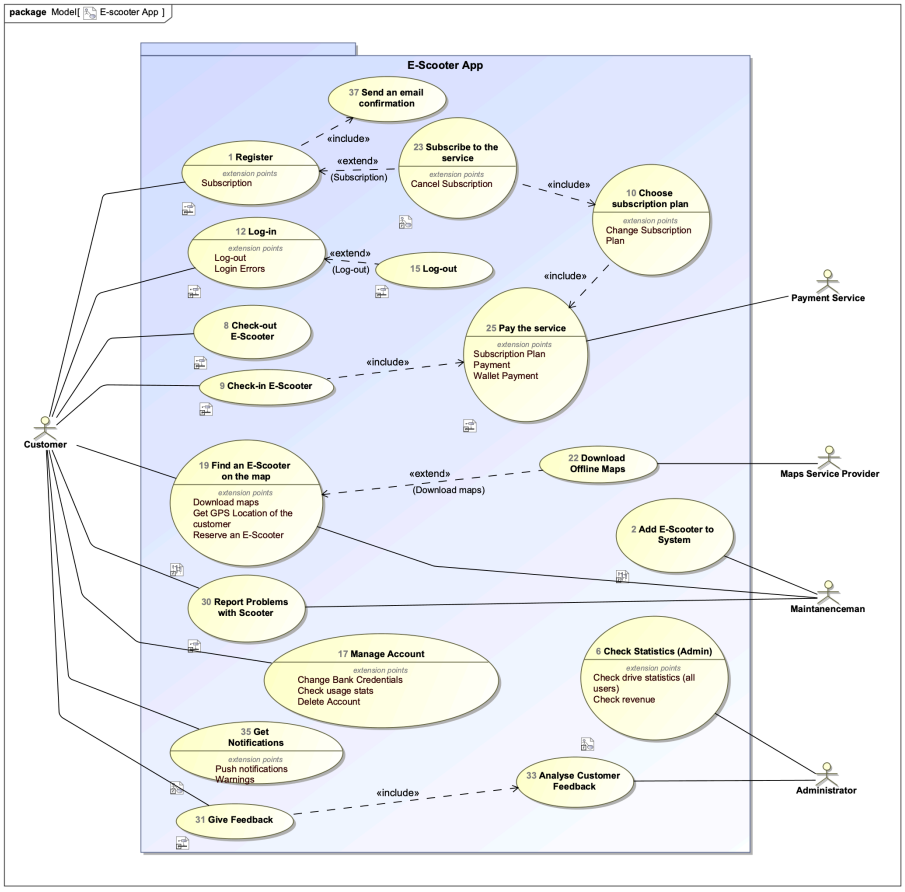
Source: https://de.wikipedia.org/wiki/LaTeX#/media/Datei:LaTeX_logo.svg

Source: https://www.brandeps.com/logo/A/Axure-01

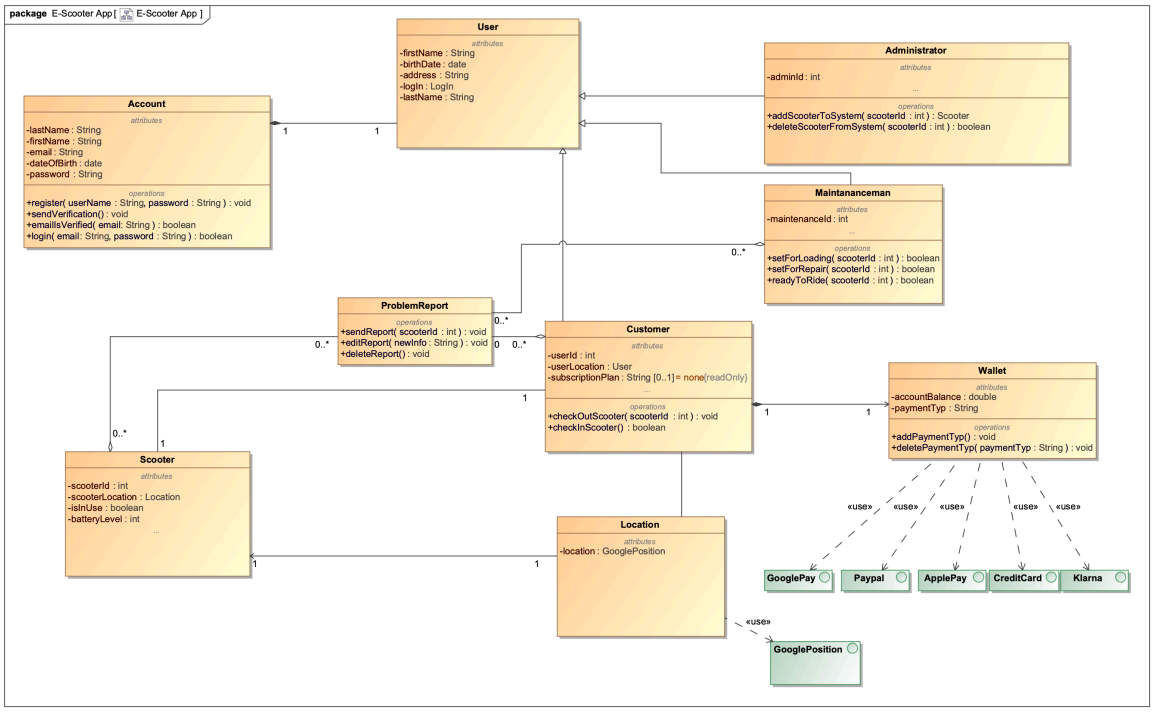
Requirements

ID	Name	Requirement Type	Associated Roles	Event/ use case #	Description	Rationale	Source	Fit Criterion	Customer Satisfaction	Customer Dissatisfaction	Dependencies	Conflicts	Supporting Materials	Priority	Estimation	Status	User Story
1	Registration	Functional	Customer	Register	Customers need to register to get personal information, payment method etc.	Customers must register to use the E-Scooter rental service	Steffen Hanzlik, Marco Peluso, Kendra Birringer	Fits if the customer can register successfully	5	5	Registration success	None		High	8	Done	As a new customer, if I want to use the service, I must be able to register and create a new account.
2	Log-in	Functional	Customer	Manage Account	Customers must log-in if they want to rent an E-Scooter	Customers must log-in to use the E-Scooter rental service	Steffen Hanzlik, Marco Peluso, Kendra Birringer	Fits if the customer can log-in	4	4	Register	None		High	5	Done	As a customer, if I want to use an E-Scooter, I must be able to log-in into my account.
3	Log-out	Functional	Customer	Manage Account	Customers must have the possibility to log-out	Customers should be able to log-out from our service	Steffen Hanzlik, Marco Peluso, Kendra Birringer	Fits if the customer can log out	3	3	Register, Log-in	None		High	2	Done	As a customer, if I finished the ride and do not need the E-Scooter anymore, I must be able to log-out of my account.
4	Change Account Details	Functional	Customer	Manage Account	Change personal information or payment	Customers should be able to change their personal information	Steffen Hanzlik, Marco Peluso, Kendra Birringer	Fits if the customer can change account details successfully	3	5	Register, Log-in	None		Medium	3	Done	As a customer I would like to be able to change my account information if something has changed, like personal information or payment method.
5	Delete Account	Functional	Customer, Admin	Manage Account	Customers and Admins can delete a customer's account	Customers should be able to delete their account if they do not want to use the E-Scooter Service any further. Also, if the customer is not following the Terms and Conditions, the Admins also should be able to remove the customer's account	Svetozar Stojanovic	Fits if the customer's account no longer exists	3	3	Register, Log-in	None		Medium	2	Done	As a customer, I want to be able to delete my account if I do not want to use the service anymore. / As an Admin I want to be able to delete a customer's account, if she/ he violates the Terms and Conditions.

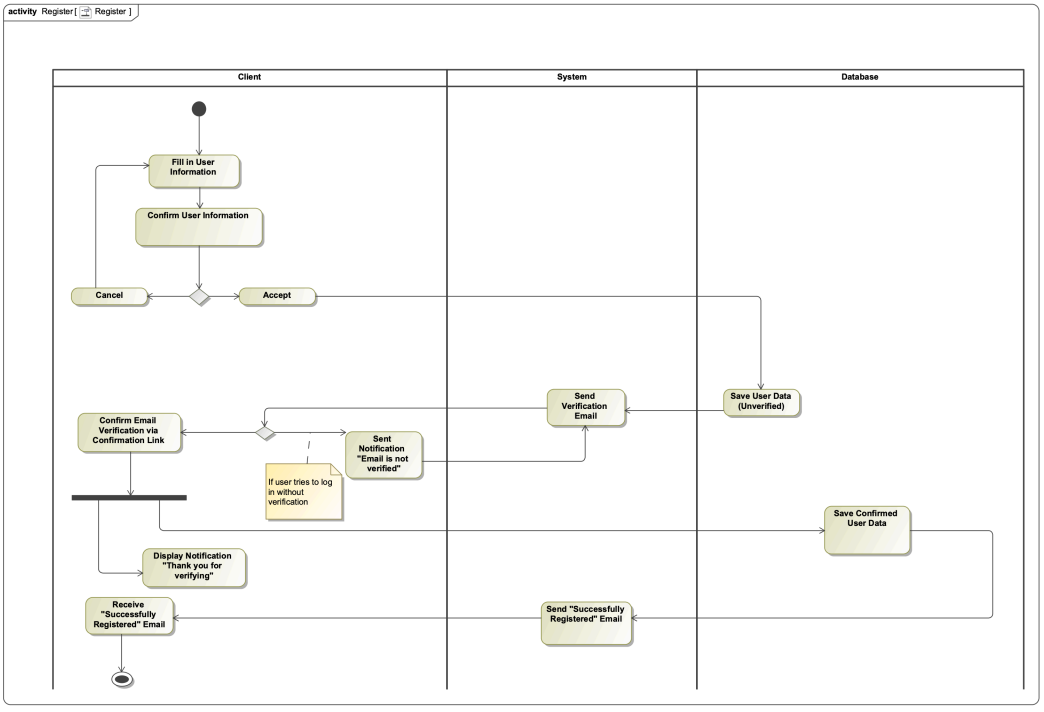
Use Case Diagram



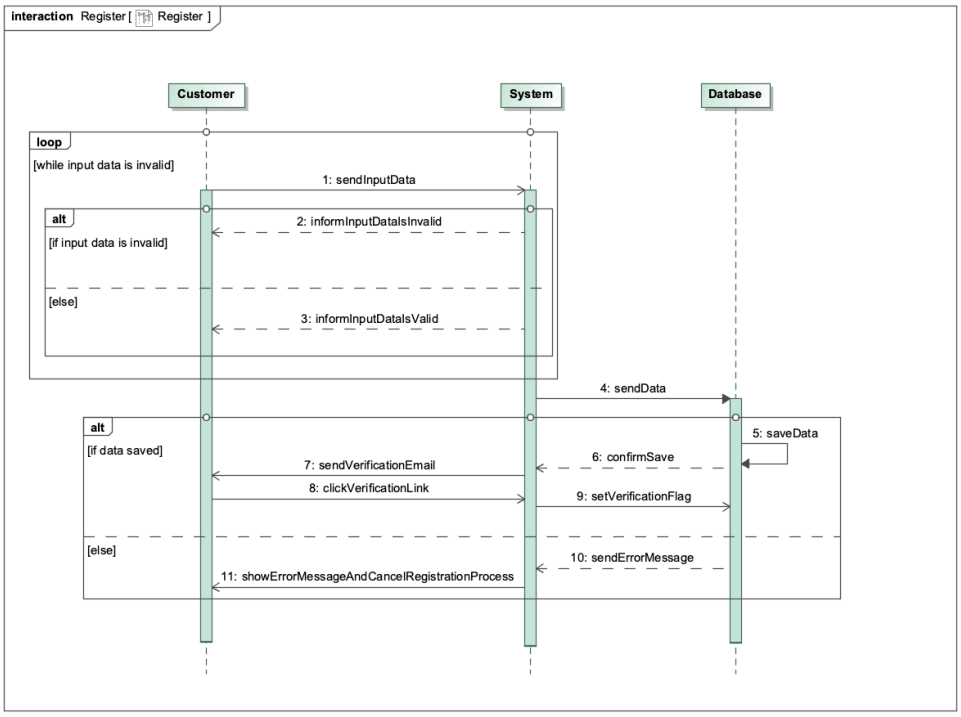
Class Diagram



Activity Diagram

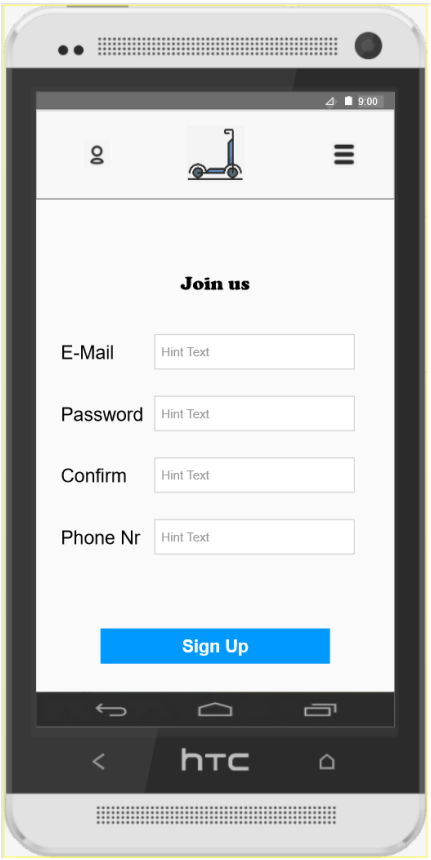


Sequence Diagram



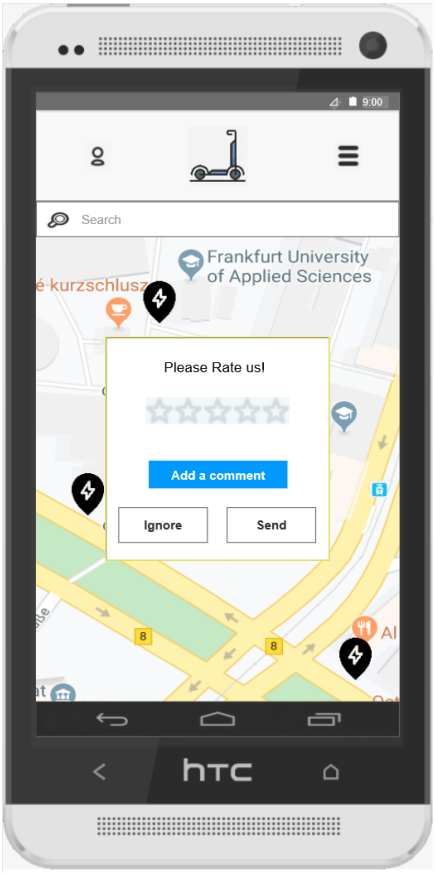
UI Prototype

- Axure Experience
  - Can be animated and used as a web app
- Focus on the mobile platform



Conclusion

- Scrum Experience
- Difficulties
  - Different time schedules for team members



11.5 UI Prototypes

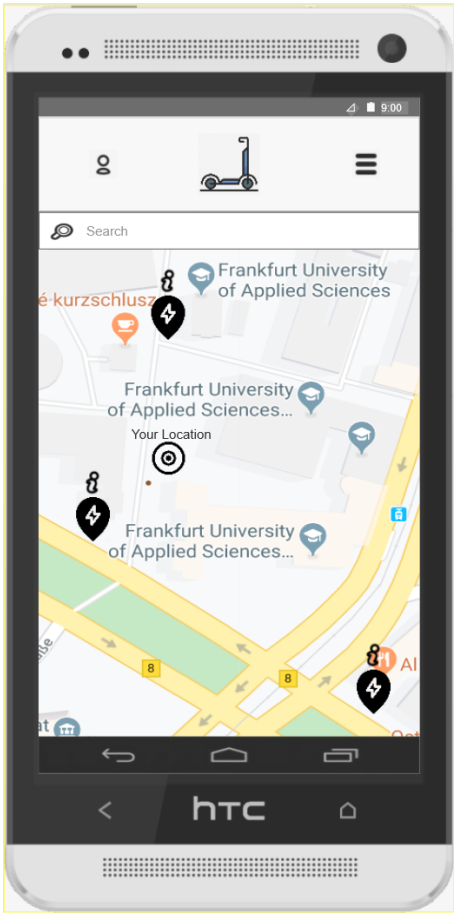


Figure 10: Start Menu

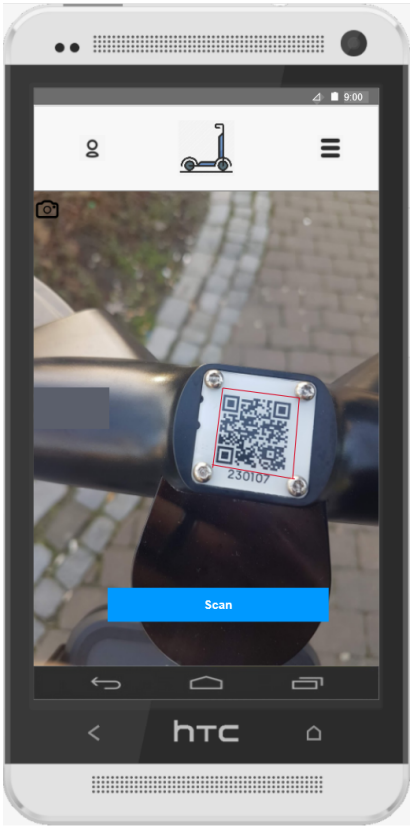


Figure 11: Start Menu → Scan QR-Code

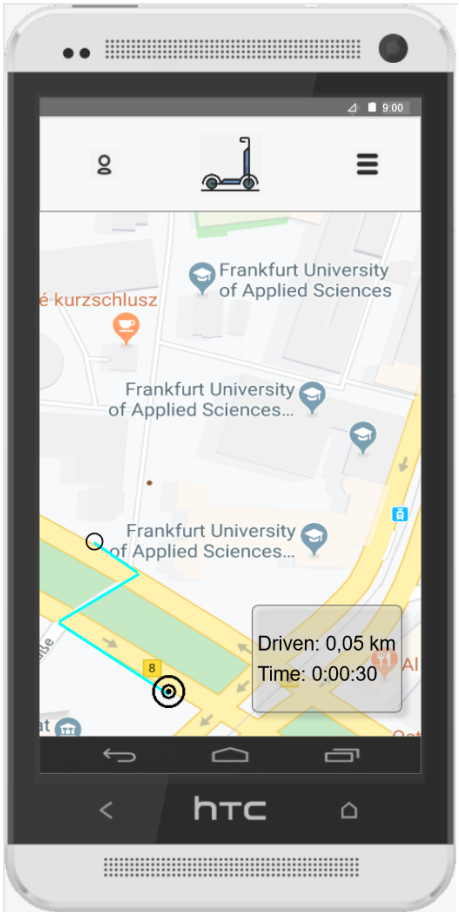


Figure 12: Start Menu → Driving

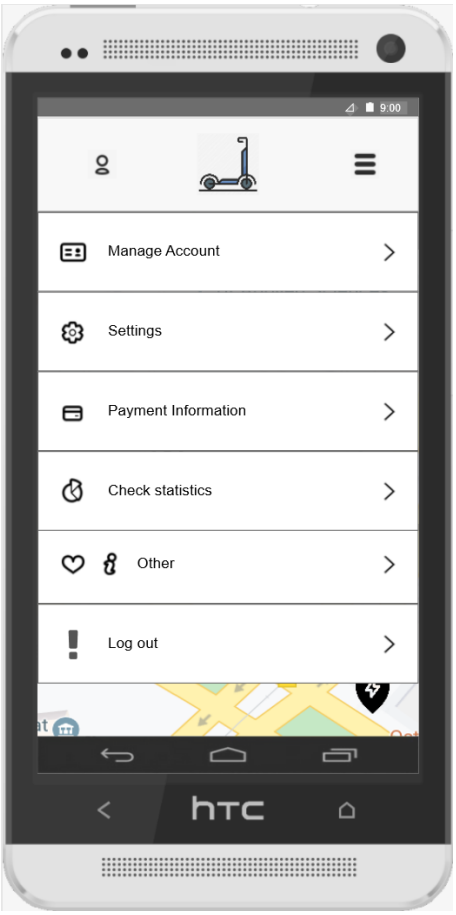


Figure 13: Menu Dropdown

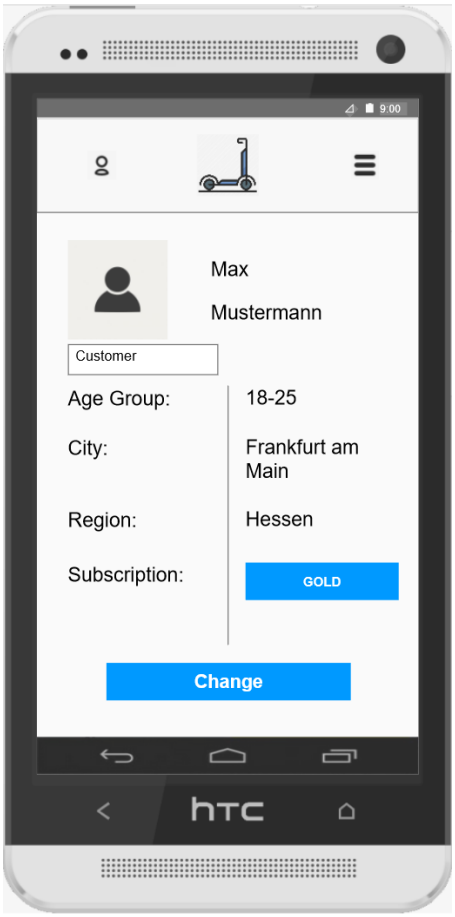


Figure 14: Menu Dropdown → Account Management

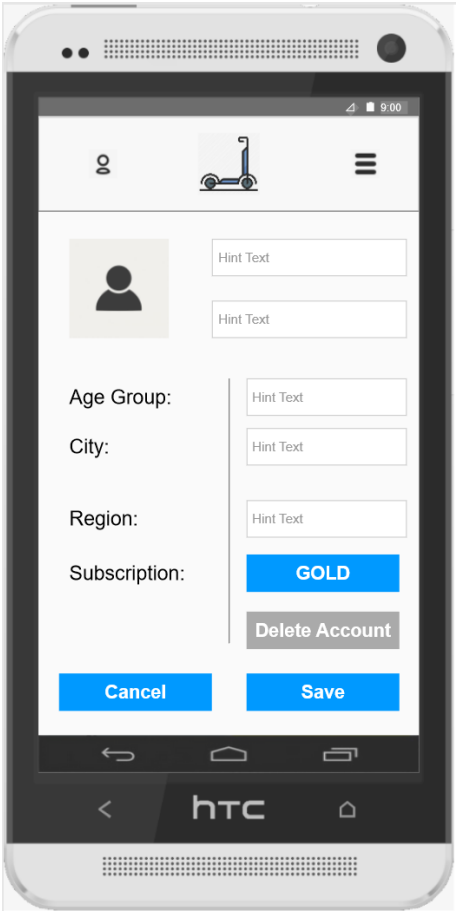


Figure 15: Menu Dropdown → Account Management → Change Account Details

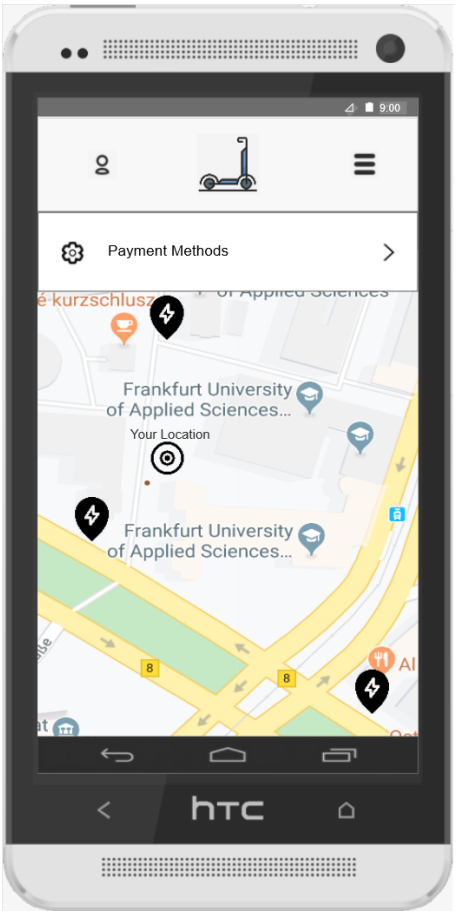


Figure 16: Menu Dropdown → Payment Information

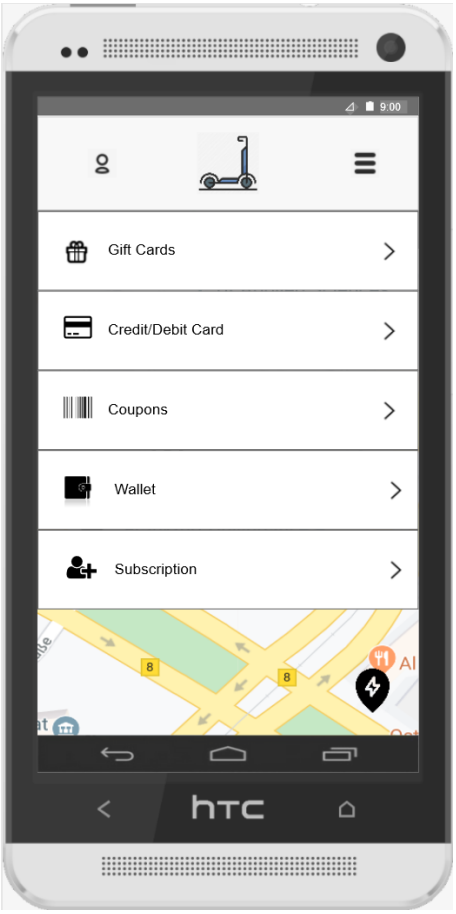


Figure 17: Menu Dropdown → Payment Information → Payment Methods

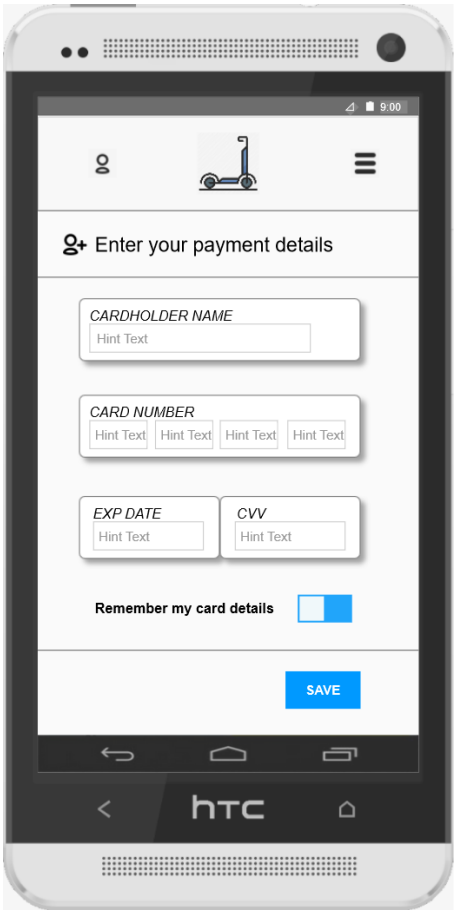


Figure 18: Menu Dropdown → Payment Information → Payment Methods → Credit/Debit Card

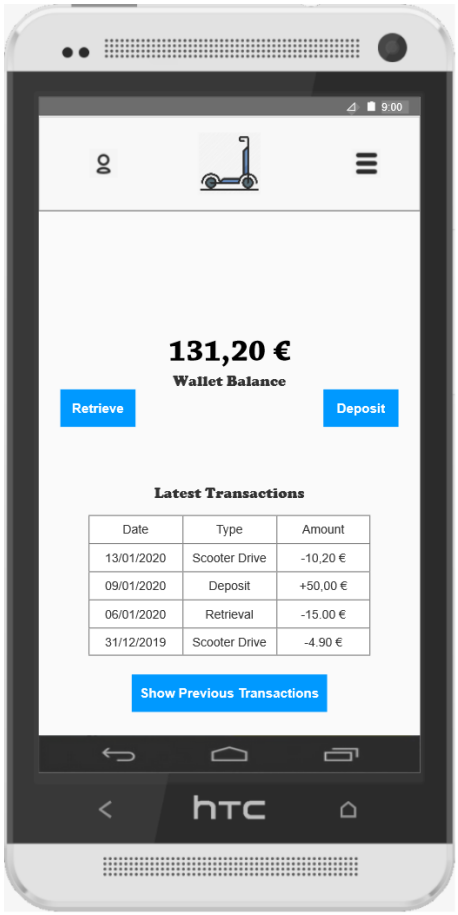


Figure 19: Menu Dropdown → Payment Information → Payment Methods → Wallet

Figure 20: Menu Dropdown → Check Drive Statistics

Figure 21: Menu Dropdown → Check Spent Money

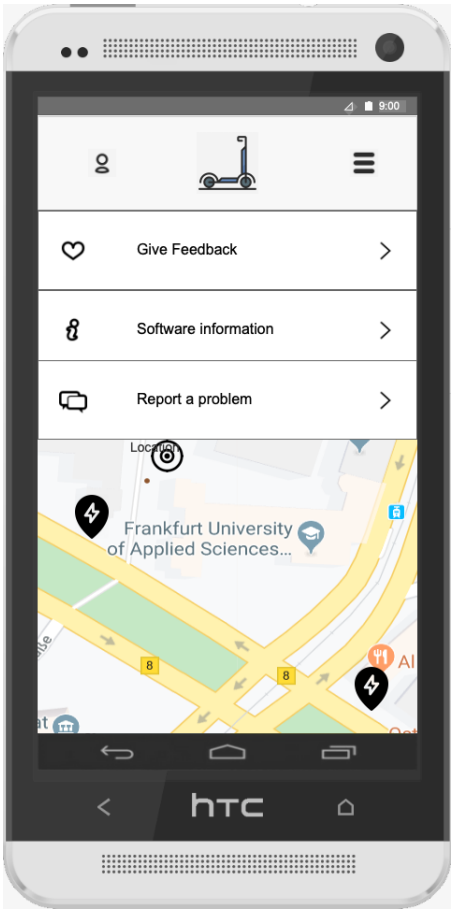


Figure 22: Menu Dropdown → Other

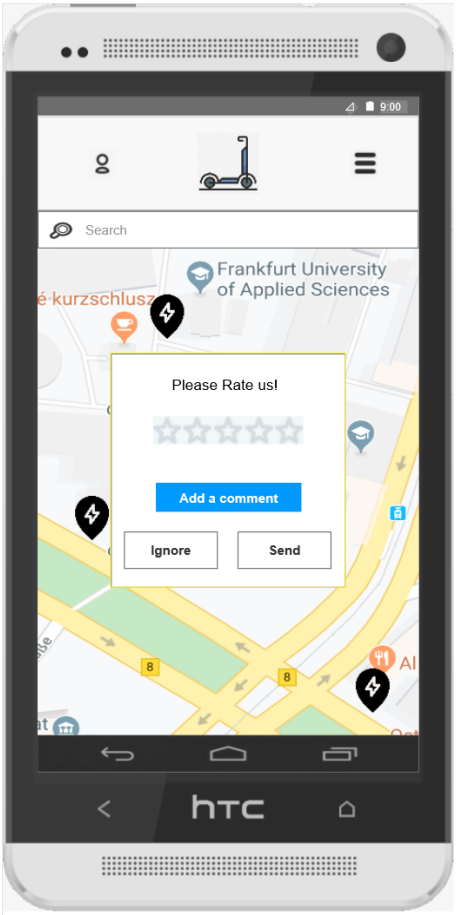


Figure 23: Ask for Feedback

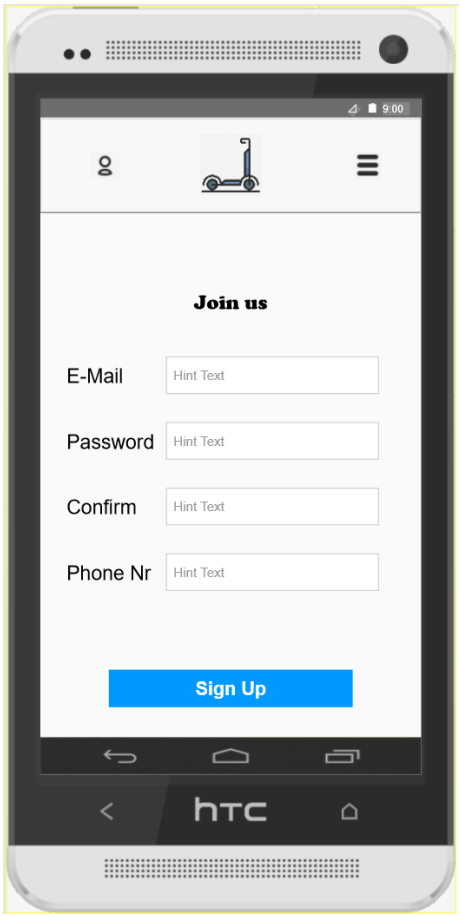


Figure 24: Register/Sign Up

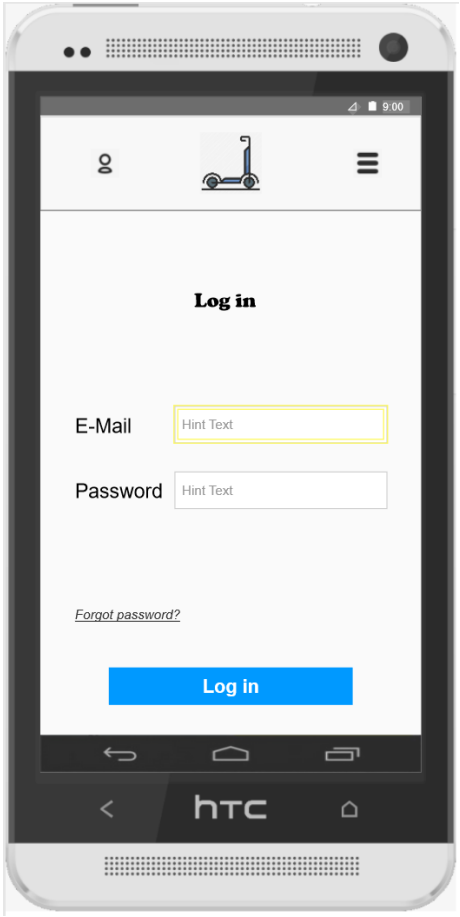


Figure 25: Log in

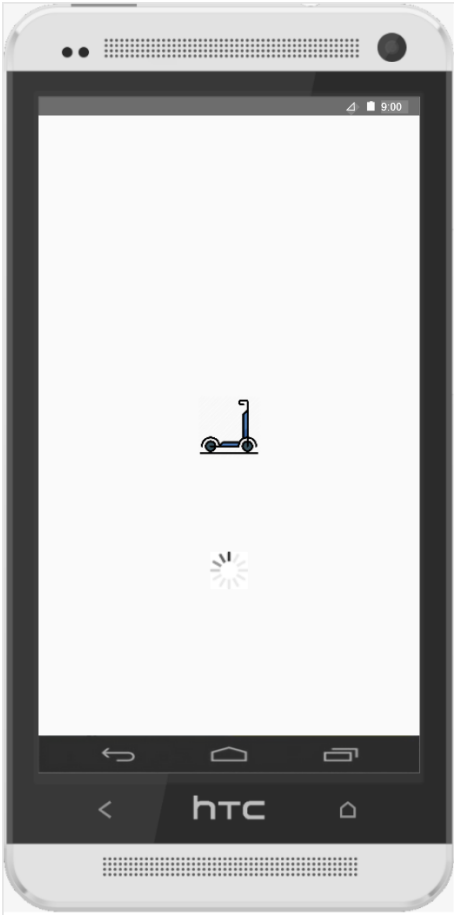


Figure 26: Loading Screen

11.6 Meeting Protocols

Weekly Scrum (Discord):

Week:	1
Date:	2019-12-22
Time:	11:00 - 12:50
Participants:	Kendra, Nader, Steffen, Marco, Svetozar

Topics:

- talked about general scope of the project
- reviewed some general questions regarding requirements, use cases, prototyping
- split up work for the coming week
- discussed finding a time slot for daily Scrum meetings, decided to do them weekly instead
- confer about structure of the overall system (web-client, app-client, server, ...)
- Nader was not sure about requirement #26
- talked about possible roles in our team
- intro to git for the less experienced team members
- played some planning poker

Team Meeting (Café 1):

Week:	1
Date:	2019-12-25
Time:	13:15 - 14:30
Participants:	Kendra, Steffen, Marco

Topics:

- set up TexMaker
- checked out Overleaf
- began to work on LaTeX documentation
- reviewed git again

Weekly Scrum (Discord):

Week:	2
Date:	2020-01-19
Time:	11:00 - 12:15
Participants:	Kendra, Nader, Steffen, Marco, Svetozar

Topics:

- discussed collected requirements and importance of individual items
- shared thoughts about the different kind of uml diagrams
- decided to do documentation in LaTeX
- split up work into different parts
- discuss clean up of big use case diagram
- reviewed some of the artifacts according to the DoD
- talked about maybe using Trello as Todo-list tool
- played some more planning poker, to improve estimations

Weekly Scrum (Discord):

Week:	3
Date:	2020-01-26
Time:	11:00 - 12:30
Participants:	Kendra, Nader, Steffen, Marco, Svetozar

Topics:

- reviewed last weeks progress
- plan for next week
- talked through several backlog items
- planning poker again
- discussed possible tools for prototyping - Svetozar wanted to try Axure
- reviewed some git commands
- decided to ditch Trello completely and instead mark backlog items todo status in Google Sheets
- talked a lot about uml diagrams

Team Meeting (Café 1):

Week:	3
Date:	2020-01-27
Time:	13:15 - 15:30
Participants:	Kendra, Steffen, Marco, Svetozar

Topics:

- tried out a few mockup / prototyping tools (draw.io, balsamiq)
- decided to stay with our decision and use Axure

Weekly Scrum (Discord):

Week:	4
Date:	2020-02-02
Time:	11:00 - 12:10
Participants:	Kendra, Nader, Steffen, Marco, Svetozar

Topics:

- talked about possible improvements of our processes
- went over requirements again
- reviewed some of the diagrams together and talked about possible improvements
- talked about the ui prototypes we wanted to build
- set goal to finish everything until end of week, so that we can focus on presentation next week

Weekly Scrum (Discord):

Week:	5
Date:	2020-02-09
Time:	11:00 - 12:10
Participants:	Kendra, Nader, Steffen, Marco, Svetozar

Topics:

- reviewed everything we had done so far
- discussed about the way we wanted to do the presentation
- realized we still have a lot of minor corrections to do in MagicDraw as well as to the documentation, before we could work on presentation - last weeks goal not really reached

Presentation Practice Meeting:

Week:	5
Date:	2020-02-07
Time:	11:00 - 12:10
Participants:	Kendra, Nader, Steffen, Marco, Svetozar

Topics:

- rehearsed presentation
- realized we needed to drastically shorten the presentation
- cut a lot of stuff out - decided to give a short overview of everything and then concentrate on a single use case, explaining its different diagrams
- rehearsed again a few times and tweaked till we were satisfied with the presentation

List of Figures

References

[1] https://www.scrum.org/resources/what-is-scrum

[2] https://www.omg.org/spec/UML/

[3] https://www.nomagic.com/products/magicdraw

[4] Prof. Dr.-Ing Peter Thoma *05 Software Engineering Analysis (Software Models and UML)*

[5] https://www.axure.com/

[6] https://www.scrumguides.org/scrum-guide.html

[7] Prof. Dr.-Ing Peter Thoma *02-3 Software Engineering Analysis (Scrum)*

[8] http://www11.informatik.uni-erlangen.de/Lehre/SS2015/PR-SWE/Material/volere-template.pdf

[9] https://www.mountaingoatsoftware.com/agile/planning-poker

[10] https://discordapp.com/