

Story	Description	Task	Description	Requirement	Description	Class Level-1	Class Level-2	Non-Functional	Non-Functional Subcategory
Story A1 "Agieren auf Plattform"	"Als Akteur möchte ich in meiner Rolle als [X] auf der Plattform agieren."	Task A1.1	Akteure können sich an der Plattform registrieren und gemäß ihrer Rolle miteinander agieren.	Requirement A1.1.1	Jeder Akteur auf der Plattform kann eindeutig identifiziert werden.	Software	Nonfunctional	Security	Accountability
				Requirement A1.1.2	Ein Akteur registriert sich und meldet sich auf der Plattform an, bevor er dort agieren kann.	Software	Functional		
				Requirement A1.1.3	Ein Akteur agiert immer mit einer bestimmten Rolle auf der Plattform: Manufacturer, Customer, Supplier, Service-Provider oder Gerät. Ein Akteur kann mehrere Rollen haben.	Software	Functional		
				Requirement A1.1.4	Es existiert eine Oberfläche, auf die jeder Akteur Zugriff hat. Dort kann er sich registrieren und anmelden.	Software	Functional		
				Requirement A1.1.5	Ein Akteur hat eine (mehrere) verifizierte Rolle(n).	Software	Functional		
		Task A1.2	Akteure können über die Plattform miteinander kommunizieren.	Requirement A1.2.1	Akteure kommunizieren über die Plattform.	Software	Functional		
				Requirement A1.2.2	Die Kommunikation der beteiligten Akteure wird sofort übermittelt.	Software	Nonfunctional	PerformanceEfficiency	Time behavior
				Requirement A1.2.3	Die Kommunikation zwischen den Akteuren ist nachvollziehbar und eindeutig zuordenbar.	Software	Nonfunctional	Security	Accountability
				Requirement A1.2.4	Die Kommunikation zwischen den Akteuren kann nicht gelöscht oder manipuliert werden.	Software	Nonfunctional	Security	Integrity
				Requirement A1.2.5	Akteure können nur den Inhalt ihrer eigenen Nachrichten einsehen.	Software	Nonfunctional	Security	Confidentiality
		Task A1.3	Akteure können über die Plattform Verträge miteinander abschließen.	Requirement A1.3.1	Akteure schließen Verträge über die Plattform ab.	Software	Functional		
				Requirement A1.3.2	Verträge sind rechtlich bindend.	Software	Functional		
				Requirement A1.3.3	Akteure können Verträge ablehnen oder annehmen.	Software	Functional		
				Requirement A1.3.4	Es existiert eine Oberfläche, auf die jeder Akteur Zugriff hat. Dort kann er Vertragsanfragen erstellen. Auf der Empfängerseite muss eine Oberfläche existieren, die diese Anfragen anzeigt.	Software	Functional		
				Requirement A1.3.5	Der Vertragsgegenstand kann nicht durch Dritte manipuliert werden.	Software	Nonfunctional	Security	Integrity
				Requirement A1.3.6	Akteure haben Zugriff auf alle Vertragsinformationen.	Software	Functional		
				Requirement A1.3.7	Akteure können komplexe Vertragskonstrukte umsetzen. Verträge haben einen Status. Diese können "Aktiv", "In Erzeugung" oder "Inaktiv" sein.	Software	Functional		
				Requirement A1.3.8	Akteure können ihre Verträge im Nachhinein ändern.	Software	Functional		
				Requirement A1.3.9	Akteure können nur ihre eigenen Verträge ändern.	Software	Nonfunctional	Security	Confidentiality
				Requirement A1.3.10	Eine Vertragsänderung bedarf der Zustimmung aller beteiligten Akteure.	Software	Functional		
				Requirement A1.3.11	Erbrachte Leistungen werden kostenpflichtig verrechnet.	Software	Functional		
		Task A1.4	Akteure benötigen grafische Oberflächen zum Agieren auf der Plattform.	Requirement A1.4.1	Es existiert eine Oberfläche, auf die der Akteur Zugriff hat. Dort hat er eine Übersicht über alle seiner Verträge sowie aggregierte Informationen wie Anzahl aller Verträge, Kontostand, etc. Es werden ebenfalls angebotene Verträge angezeigt, die angenommen oder abgelehnt werden können.	Software	Functional		
				Requirement A1.4.2	Es existiert eine Oberfläche, auf die der Akteur Zugriff hat. Dort hat er eine detaillierte Übersicht über einen seiner Verträge und kann sich Detailinformationen dazu ansehen. Außerdem sieht er eine Übersicht über alle Nachrichten, die mit diesem Vertrag in Verbindung stehen und den Vertrag bearbeiten.	Software	Functional		
Story M1 "Geräte vermieten"	"Als Manufacturer möchte ich meine Geräte über die Plattform vermieten können, um meinen Umsatz zu steigern."	Task M1.1	Die Geräte können auf der Plattform vermietet werden.	Requirement M1.1.1	Ein Gerät ist Eigentum eines Manufacturers.	Software	Functional		
				Requirement M1.1.1	Ein Manufacturer kann beliebig viele Geräte besitzen und über die Plattform vermieten.	Software	Functional		
		Task M1.2	Die Geräte benötigen Sensoren, um Fehler und Defekte zu detektieren.	Requirement M1.2.1	Geräte können Fehlerzustände detektieren. Tritt ein Fehler auf, wird dieser dem Customer über ein Display angezeigt.	Software	Functional		
				Requirement M1.2.2	Geräte können Defekte detektieren. Tritt ein Fehler auf, wird dieser über die Plattform an den Service-Provider (Service-Vertrag) gemeldet.	Software	Functional		
				Requirement M1.2.3	Ein Defekt (Defektes Mahlwerk, undichte Anschlüsse, etc.) hindert das Gerät am Durchführen seiner Tätigkeit und muss durch einen Service-Provider behoben werden.	Software	Functional		
				Requirement M1.2.4	Ein Fehlerzustand (Leerer Wasserbehälter, geöffnete Abdeckung, etc.) hindert das Gerät am Durchführen seiner Tätigkeit und kann meistens durch den Customer behoben werden.	Software	Functional		
Story M2 "Verträge erzeugen"	"Als Manufacturer möchte ich in der Lage sein, Verträge anzulegen, um meine Geräte über die Plattform vermieten zu können."	Task M2.1	Ein Manufacturer kann Verträge erzeugen.	Requirement M2.1.1	Es existiert eine Oberfläche, auf die der Manufacturer Zugriff hat. Dort kann er Mietverträge erzeugen und als Antwort auf seine Mietanfrage an den Customer senden.	Software	Functional		
				Requirement M2.1.2	Es existiert eine Oberfläche, auf die der Manufacturer Zugriff hat. Dort kann er alle Service-Provider und deren Dienstleistungen einsehen. Service-Verträge erzeugen und an einen Service-Provider senden.	Software	Functional		
				Requirement M2.1.3	Es existiert eine Oberfläche, auf die der Manufacturer Zugriff hat. Dort kann er alle Supplier einsehen, Lieferverträge erzeugen und an den Supplier senden.	Software	Functional		
Story M3 "Verträge abrechnen"	"Als Manufacturer benötige ich eine korrekte, nutzungabhängige und automatische Abrechnung der vermieteten Geräte, die regelmäßig aktualisiert wird, sowie der erbrachten Dienstleistungen, um den Umsatz aufrecht zu erhalten."	Task M3.1	Die Geräte müssen den Verbrauch detektieren.	Requirement M3.1.1	Geräte detektieren den Verbrauch (Kaffeemaschine: Anzahl Kaffees) und senden diesen an die Plattform.	Software	Functional		
		Task M3.2	Die Geräte müssen die Verbrauchsdaten an die Plattform melden.	Requirement M3.2.1	Geräte senden den Verbrauch nach Fertigstellung des Produktes sofort an die Plattform.	Software	Nonfunctional	PerformanceEfficiency	Time behavior
				Requirement M3.2.2	Die Verbrauchsdaten der Geräte können nicht manipuliert werden.	Software	Nonfunctional	Security	Integrity
Story C1 "Ansicht verfügbarer Geräte"	"Als Customer möchte ich verfügbare Haushaltsgeräte angezeigt bekommen, um das passende Gerät mieten zu können."	Task C1.1	Es muss eine Auflistung aller verfügbarer (mietbarer) Geräte existieren.	Requirement C1.1.1	Es existiert eine Oberfläche, auf die der Customer Zugriff hat. Dort hat er die Möglichkeit, alle verfügbaren Geräte aufzulisten.	Software	Functional		
		Task C1.2	Es müssen alle für den Customer relevanten Informationen über das Gerät vorhanden und einsehbar sein.	Requirement C1.2.1	Der Customer hat Zugriff auf eine detaillierte Beschreibung des Geräts.	Software	Functional		
		Task C2.1	Das Gerät muss die Reinigung / Wartung durch den Customer detektieren können.	Requirement C2.1.1	Geräte detektieren eine Reinigung (Produktbehälter leeren, etc.).	Software	Functional		
				Requirement C2.1.2	Geräte detektieren eine Wartung (Entkalken, etc.).	Software	Functional		

Story C2  "Geräte warten"	"Als Customer möchte ich die gemieteten Geräte reinigen und warten können, um dafür vom Hersteller eine Gutschrift auf mein Vertragskonto zu erhalten."	Task C2.2	Es muss sichergestellt werden, dass eine Reinigung / Reparatur dem Gerät bzw. dessen Sensoren nicht vorgespült werden kann.	Requirement C2.2.1	Detektierte Reinigungen / Wartungen können nicht manipuliert oder dem Gerät vorgespült werden.	Software	Nonfunctional	Security	Integrity
		Task C2.3	Der Customer muss vom Gerät eindeutig identifiziert werden können.	Requirement C2.3.1	Das Gerät kann den Customer identifizieren.	Software	Functional		
		Task C2.4	Die Reinigung / Wartung muss an die Plattform übertragen werden und gemäß des Vertrages abgerechnet werden.	Requirement C2.4.1	Das Gerät sendet detektierte Reinigungen / Wartungen an die Plattform.	Software	Functional		
				Requirement C2.4.2	Das Gerät sendet detektierte Reinigungen / Wartungen nach Abschluss direkt an die Plattform.	Software	Nonfunctional	PerformanceEfficiency	Time behavior
Story C3  "Einfache Bedienbarkeit der Plattform"	"Als Customer möchte ich eine intuitive, einfach zu bedienende Oberfläche, um mich gut auf der Plattform zurechtzufinden."	Task C3.1	Ein Customer muss zunächst mit der Plattform bekannt gemacht werden.	Requirement C3.1.1	Ein Customer erhält eine initiale Einführung über die Plattform bei der ersten Anmeldung.	System	Transition		
				Requirement C3.1.2	Die Funktionen der Plattform sind für den Customer schnell erlernbar und leicht verständlich.	Software	Nonfunctional	Usability	Learnability
		Task C3.2	Die Plattform muss optisch ansprechend sein.	Requirement C3.2.1	Die Plattform ist für den Customer optisch ansprechend.	Software	Nonfunctional	Usability	User interface aesthetics
Story SP1  "Dienstleistungen bereitstellen"	"Als Service-Provider möchte ich meine Angebotspalette auf der Plattform anbieten können."	Task SP1.1	Ein Service-Provider muss seine angebotenen Dienstleistungen auf der Plattform eingeben können.	Requirement SP1.1.1	Es existiert eine Oberfläche, auf die der Service-Provider Zugriff hat. Dort kann er alle Dienstleistungen, die er anbietet, sowie Detailinformationen, wie zum Beispiel Kosten der Dienstleistung, eintragen und damit auf der Plattform verfügbar machen.	Software	Functional		
				Requirement SP1.1.2	Es existiert eine Oberfläche, auf die der Service-Provider Zugriff hat. Dort kann er bereits angebotene Dienstleistungen editieren oder löschen.	Software	Functional		
Story SP2  "Service-Aufträge abschließen"	"Als Service-Provider möchte ich nach der Durchführung der Wartung diese mit meinem Smartphone am Gerät bestätigen, um den Service-Auftrag abzuschließen."	Task SP2.1	Der Service-Provider kann mit dem Gerät per Smartphone kontaktlos kommunizieren.	Requirement SP2.1.1	Gerät und Service-Provider können miteinander kommunizieren.	Software	Functional		
				Requirement SP2.1.2	Ein Gerät kann den Service-Provider eindeutig identifizieren.	Software	Functional		
		Task SP2.2	Es wird eine App benötigt, mit der der Service-Provider seine Identität und die durchgeführte Wartung am Gerät bestätigen kann.	Requirement SP2.2.1	Es existiert eine Smartphone-Oberfläche, auf die der Service-Provider Zugriff hat. Darüber kann er Wartungen abschließen.	Software	Functional		
				Requirement SP2.2.2	Ein Service-Provider kann einen Service-Auftrag starten und beenden, wenn er sich in der Nähe des Gerät befindet.	Software	Functional		
Story SEC1  "Sichere Zahlungsabwicklung"	"Als IT-Security-Beauftragter möchte ich sichergestellt wissen, dass die Zahlungsabwicklung auf der Plattform sicher und voll funktionsfähig ist."	Task SEC1.1	Es muss sichergestellt werden, dass Geldtransfers vom Sender an den Empfänger durchgeführt werden.	Requirement SEC1.1.1	Geldtransfers werden auf der Plattform geloggt.	Software	Functional		
				Requirement SEC1.1.2	Bei Nicht-Ausführung von Geld- und Nachrichtentransfers werden die Parteien benachrichtigt.	Software	Nonfunctional	Security	Integrity
				Requirement SEC1.1.3	Die Plattform prüft Geldtransfers vor Ausführung.	Software	Functional		
		Task SEC1.2	Es muss sichergestellt werden, dass Sender und Empfänger des Geldes eindeutig identifizierbar sind.	Requirement SEC1.2.1	Jeder Akteur besitzt eine (mehrere) eindeutige Kontonummer(n).	Software	Nonfunctional	Security	Accountability
				Requirement SEC1.2.2	Konten sind zugriffsgeschützt.	Software	Nonfunctional	Security	Authenticity
Story SEC2  "Sichere Kommunikation und signierte Nachrichten"	"Als IT-Security-Beauftragter möchte ich eine verschlüsselte Kommunikation mit der Plattform, damit meine Daten nicht in die Hände von Dritten gelangen."	Task SEC2.1	Die Kommunikation zwischen Akteuren und der Plattform muss verschlüsselt werden.	Requirement SEC2.1.1	Sämtliche Verbindungen sind per SSL/TLS zu verschlüsseln.	System	Quality		
		Task SEC2.2	Aktuelle Sicherheitsstandards müssen verwendet werden.	Requirement SEC2.2.1	Es können Passwortregeln hinterlegt werden. Die Einhaltung dieser Regeln wird überprüft.	System	Quality		
Story SEC3  "Manipulationssicherheit"	"Als IT-Security-Beauftragter möchte ich eine manipulationssichere Plattform, um die Integrität und Echtheit der Daten zu gewährleisten."	Task SEC3.1	Es müssen Vorkehrungen gegen Manipulationen getroffen werden.	Requirement SEC3.1.1	Der Zugang zu den Backend-Systemen wird protokolliert und nur Berechtigten gestattet.	Software	Nonfunctional	Security	Accountability
				Requirement SEC3.1.2	Manipulationen werden durch den Einsatz kryptographischer Methoden verhindert.	Software	Nonfunctional	Security	Integrity
		Task SEC3.2	Es ist nachvollziehbar, wer wann auf die Plattform zugegriffen hat.	Requirement SEC3.2.1	Alle Aktivitäten auf der Plattform werden geloggt.	Software	Nonfunctional	Security	Accountability
Story BD1  "Geschäftsmodell"	"Als Business-Developer möchte ich mit dem Geschäftsmodell Pay-As-You-Use einen größeren Kundenbereich ansprechen und den Umsatz steigern."	Task BD1.1	Das Pay-As-You-Use Abrechnungsmodell muss in einem Vertrag abgebildet werden können.	Requirement BD1.1.1	Vermietete Geräte werden nach dem Pay-As-You-Use Prinzip abgerechnet.	System	Business		
		Task BD1.2	Es ist vertraglich festgelegt, welcher Akteur für welche Dienstleistung wie viel Geld bezahlt bzw. erhält.	Requirement BD1.2.1	Customer bezahlen jede verbrauchte Einheit (z.B. pro Tasse Kaffee) des gemieteten Geräts an den Manufacturer. Die genauen Kosten sind vom Gerät abhängig und werden durch den Manufacturer festgelegt.	Software	Functional		
				Requirement BD1.2.2	Manufacturer bezahlen Customer für jede durchgeführte Wartung des vermieteten Geräts. Die Höhe der Zahlung ist im Mietvertrag geregelt.	Software	Functional		
				Requirement BD1.2.3	Manufacturer bezahlen Service-Provider für jede erbrachte Service-Leistung. Die Höhe der Zahlung ist im Service-Vertrag geregelt.	Software	Functional		
				Requirement BD1.2.4	Verträge sind individuell gestaltbar.	Software	Functional		
				Requirement BD1.2.5	Manufacturer bezahlen Supplier für jede erbrachte Lieferung. Die Höhe der Zahlung ist im Liefervertrag geregelt.	Software	Functional		
Story BD2  "Plattform für Partner"	"Als Business-Developer möchte ich eine Plattform, die für Partner wie Service-Provider oder Supplier leicht zugänglich ist."	Task BD2.1	Partner agieren auf der Plattform.	Requirement BD2.1.1	Auf der Plattform agieren Geschäftspartner (Service-Provider, Supplier, ect.).	System	Business		
				Requirement BD2.1.2	Die Nutzung der Plattform ist intuitiv und schnell erlernbar. Umfangreiche Mitarbeiterschulungen zur Benutzung der Plattform sind nicht notwendig.	System	Transition		
		Task BD2.2	Partner müssen als solche identifiziert sein.	Requirement BD2.2.1	Der Prozess zur Prüfung, dass es sich z.B. bei einem Service-Provider auch tatsächlich um einen solchen handelt, ist benutzerfreundlich und so schnell und einfach wie möglich umsetzbar.	System	Stakeholder		
Story BD3  "Hersteller-übergreifende Plattform"	"Als Business-Developer möchte ich in Zukunft eine Plattform schaffen, auf der Hersteller unterschiedlicher Branchen ihre Produkte nach dem Pay-As-You-Use Prinzip vermieten können, um dem Kunden eine breitere Produktpalette zu bieten."	Task BD3.1	Die Plattform muss zukünftig weitere Hersteller zulassen, um ein ganzes Ökosystem von Geräten aller Art dem Customer zugänglich zu machen.	Requirement BD3.1.1	Es können in Zukunft weitere Hersteller auf der Plattform ihre Geräte zur Miete anbieten.	System	Business		
		Task BD3.2	Ein Gerät muss auf der Plattform generisch repräsentiert werden und darf nicht von einem bestimmten Produkttyp oder Hersteller abhängig sein.	Requirement BD3.2.1	Die Plattform kann Geräte unterschiedlicher Art anbinden und abrechnen.	Software	Nonfunctional	Compatibility	Interoperability
		Task BD3.3	Ein Vertrag muss auf der Plattform generisch repräsentiert werden und darf nicht von einem bestimmten Produkttyp oder Hersteller abhängig sein.	Requirement BD3.3.1	Die Plattform kann unterschiedliche Vertragsarten abbilden und abrechnen.	Software	Nonfunctional	Compatibility	Interoperability
Story BD4  "Vertragsgestaltung"	"Als Business-Developer möchte ich die Plattform dazu nutzen, künftig andere Vertragsarten umzusetzen, um weitere Geschäftsfelder und Kunden zu gewinnen."	Task BD4.1	Die Einbindung von Verträgen sowie deren Struktur muss möglichst modular und unabhängig geschehen, damit später andere Verträge leicht integriert werden können.	Requirement BD4.2.1	Ein Abrechnungsmodell wird in einem Vertrag abgebildet.	System	Business		
				Requirement BD4.2.2	Einzelne Bestandteile der Plattform beeinflussen sich gegenseitig nicht und können leicht ausgetauscht werden.	Software	Nonfunctional	Maintainability	Modularity
Story SA1  "Modularer Aufbau"	"Als System-Architekt möchte ich eine modular aufgebaute Plattform, damit diese später..."	Task SA1.1	Funktionalitäten werden in Modulen gekapselt und als Service bereitgestellt.	Requirement SA1.1.1	Ein Software-Modul wird unabhängig von anderen Modulen bereitgestellt und gewartet.	Software	Process		
Story SA2	"Als System-Architekt möchte ich SPOFs und Datenverlust vermeiden."	Task SA2.1	Die Plattform kann im Falle eines Crashes alle Daten konsistent halten.	Requirement SA2.1.1	Bei Ausfall einzelner Komponenten gehen keine Daten verloren.	Software	Nonfunctional	Security	Integrity
				Requirement SA2.1.2	Bei Ausfall einzelner Komponenten entstehen keine Dateninkonsistenzen.	Software	Nonfunctional	Security	Integrity

"Redundanz"	damit das System auch im Fehlerfall weiter funktionsfähig ist.	Task SA2.2	Die Plattform darf keinen SPOF haben, damit im Falle eines Crashes das Gesamtsystem weiterhin lauffähig ist.	Requirement SA2.2.1	Die Plattform ist in der Lage, die Kommunikation und Datenverarbeitung bei bis zu 10.000 Endgeräten durchzuführen.	Software	Nonfunctional	Reliability	Availability
				Requirement SA2.2.2	Die Plattform ist nicht von einer einzelnen Komponente (SPOF) abhängig.	Software	Nonfunctional	Reliability	Availability
Story P1 "Deployment & Testing"	"Als Betreiber der Plattform möchte ich eine automatisierte und korrekte Bereitstellung der Plattform, um meinen Aufwand zu reduzieren."	Task P1.1	Es wird nur getestete Software bereitgestellt.	Requirement P1.1.1	Für alle Software-Komponenten existieren hinreichende Tests, die Test-Abdeckung beträgt 80%.	Software	Nonfunctional	Maintainability	Testability
		Task P1.2	Die Bereitstellung ist automatisiert.	Requirement P1.2.1	Die Software wird automatisiert durch Skripte installiert und bereitgestellt.	Software	Nonfunctional	Portability	Installability