

Solvas Fleet - Use Cases

Steven Bastiaens
Benjamin Cousaert
Nils Mak
Niko Strijbol
Karel Vandenbussche
David Vandorpe
Domien Van Steendam

Contents

Klant	2
Use case: Opvragen van factuur van de polis door klant	2
Use case: Raadplegen van gegevens (van polis) door klant	3
Use case: Klanten oplist	3
Use case: Klant toevoegen	4
Use case: Klant verwijderen	4
Use case: Klant wijzigen	5
Use case: Opvragen van de premie van een specifieke verzekeraar door klant	6
Use case: Opvragen premie van een subvloot door klant	6
Use case: Opvragen premie van een vloot door klant	7
Use case: Opvragen premie van een voertuig door klant	8
Use case: Simuleren van de premiewijzigingen bij het veranderen van verzekering en/of toevoegen van voertuigen	8
Use case: Wijzigen van wachtwoord van niet-ingelogde klant	9
Use case: Aanmaken van een nieuwe medewerker of gebruiker voor bank/leasing of verzekeraar	10
Use case: Aanmaken van een nieuwe gebruiker voor een klant	11
Use case: Verwijderen van een gebruiker	12
Use case: Wijzigen van wachtwoord van niet-ingelogde klant	13
Use case: Een gebruiker wijzigt zijn eigen e-mail	14
Use case: Gegevens wijzigen van een medewerker	15
Use case: Gegevens wijzigen van een gebruiker van een klant	15
Use case: Wijzigen van de rechten van een gebruiker	16
Use case: Wijzigen van wachtwoord van ingelogde klant	17
Voertuig	18

Use case: Voertuigen van een subvloot oplijsten	18
Use case: Voertuig aan een subvloot toevoegen	19
Use case: Voertuig verwijderen van een subvloot	20
Use case: Voertuig wijzigen	21
Subvloot	22
Use case: Voertuigen van een subvloot van een klant oplijsten	22
Use case: Subvloot van een klant toevoegen	22
Use case: Subvloot van een klant wijzigen	23
Use case: Subvloot van een klant wijzigen	23
Verzekeringsmaatschappij	24
Use case: Facturen van een verzekeringsmaatschappij opvragen	24
Use case: Nog niet betaalde facturen van een verzekeringsmaatschappij bekijken	25
Extra	26
Use case: Importeren van Data	26
Use case: Exporteren van Data	26
Use case: afsluiten van een verzekering	27

Klant

Use case: Opvragen van factuur van de polis door klant

Primaire actor: De klant (een bedrijf)

Objectief: De klant moet in staat zijn om elke factuur van zijn polis raad te plegen

Normale flow:

1. De gebruiker selecteert de datum van de factuur die hij wil raadplegen
2. Het systeem haalt de factuur van de gebruiker op de gekozen datum uit het archief en geeft deze weer aan de gebruiker

Pre-condities: De klant is ingelogd op de webapplicatie

Post-condities: De klant heeft zijn factuur ontvangen

Alternatieve flow: * (1). De gebruiker selecteert geen datum

- a. Het systeem vult de huidige datum in
 - b. Het systeem berekent de nieuwe factuur voor de huidige periode en geeft deze weer
- (1). De gebruiker selecteert de huidige datum

- a. Het systeem berekent de nieuwe factuur voor de huidige periode en geeft deze weer

Use case: Raadplegen van gegevens (van polis) door klant

Primaire actor: De klant (een bedrijf)

Objectief: De klant moet in staat zijn om alle nodige informatie over zijn polis op een bepaalde datum te bekijken

Normale flow:

1. De gebruiker selecteert de datum van de gegevens die hij wil raadplegen
2. Het systeem geeft de gegevens weer van de gebruiker op de gekozen datum

Pre-condities: De klant is ingelogd op de webapplicatie

Post-condities: De klant heeft zijn gegevens geraadpleegd

Alternatieve flow:

1. De gebruiker selecteert geen datum
 - a. Het systeem vult de huidige datum in
 - b. Ga naar (2)

Use case: Klanten ophijsten

Primaire actor: Een gebruiker

Objectief: De gebruiker met de juiste rechten moet klanten kunnen ophijsten

Normale flow:

1. De gebruiker gaat op de webapplicatie naar klanten ophijsten
2. (optioneel) De gebruiker geeft filters en sorteervormen door
3. De gebruiker bevestigt de zoekopdracht
4. De gebruiker krijgt de lijst te zien op de webapplicatie

Pre-condities: - De gebruiker is ingelogd op de webapplicatie - De gebruiker heeft rechten om klanten te bekijken (als verzekeringsmaatschappij of medewerker van Solvas)

Post-condities: /

Use case: Klant toevoegen

Primaire actor: Productie- of schadebeheerder

Objectief: De productie- of schadebeheerder moet een klant kunnen toevoegen aan de databank

Normale flow:

- (1. Het bedrijf komt bij Solvas met de intentie om klant te worden)
- (2. De klant geeft de nodige informatie door aan de medewerker van Solvas)
 3. De medewerker laat het systeem weten dat hij/zij een nieuwe klant wil toevoegen.
 4. De medewerker voert de gegevens in
 5. Het systeem vraagt om bevestiging
 6. De medewerker bevestigt de toevoeging en slaat de klant persistent op.

Pre-condities: - De medewerker is ingelogd op de webapplicatie en heeft de nodige rechten - De klant is nog niet gekend bij het systeem

Post-condities: De klant werd toegevoegd aan de databank

Alternatieve flow * (5). Gegevens staan in een fout formaat of zijn onvolledig
a. Het systeem toont een foutmelding met mogelijkheid om gegevens aan te passen

b. Ga naar 5

- (5). De medewerker annuleert de toevoeging
 - a. De medewerker bedenkt zich en drukt op annuleren

Exceptionele flow

- (6). De klant kan niet opgeslaan worden om ongekennde reden.
Het systeem toont een foutmelding met de oorzaak.

Use case: Klant verwijderen

Primaire actor: Productie- of schadebeheerder

Objectief: De productie- of schadebeheerder moet een klant kunnen verwijderen

Normale flow: 1. De medewerker van Solvas zoekt de klant op

2. De medewerker selecteert de klant en geeft aan dat hij de klant wil verwijderen
3. Het systeem vraagt om bevestiging

4. De medewerker bevestigt de verwijdering

Pre-condities: - De medewerker is ingelogd op de webapplicatie en heeft de nodige rechten - De klant staat in het systeem - De klant heeft geen lopende verzekeringen

Post-condities: - De gegevens van de klant werden gewijzigd - De gebruikers van de klant worden na een vaste periode ook gedeactiveerd, zodat ze geen toegang meer krijgen tot de webapplicatie.

Alternatieve flow * (4). De medewerker annuleert de verwijdering

- a. De medewerker bedenkt zich en drukt op annuleren

Use case: Klant wijzigen

Primaire actor: Productie- of schadebeheerder

Objectief: De productie- of schadebeheerder moet de gegevens van een klant kunnen wijzigen

Normale flow:

1. De medewerker van Solvas zoekt de klant op
2. De medewerker selecteert de klant en geeft aan dat hij de gegevens wilt veranderen
3. De medewerker vult de nieuwe gegevens in
4. Het systeem vraagt om bevestiging
5. De medewerker bevestigt de wijzigingen

Pre-condities: - De medewerker is ingelogd op de webapplicatie en heeft de nodige rechten - De klant staat in het systeem

Post-condities: - De gegevens van de klant werden gewijzigd - De historiek moet nog steeds kloppen met deze nieuwe wijzigingen.

Alternatieve flow * (4). Gegevens staan in een fout formaat a. Het systeem toont een foutmelding met mogelijkheid om gegevens aan te passen

- b. Ga naar 4

- (5). De medewerker annuleert de wijzigingen

- a. De medewerker bedenkt zich en drukt op annuleren

Use case: Opvragen van de premie van een specifieke verzekeraar door klant

Primaire actor: De klant (een bedrijf)

Objectief: De klant moet in staat zijn om de verschillende premies van zijn verzekeraars te raadplegen

Normale flow:

1. De gebruiker selecteert de verzekeraar waarvan hij de premie wil krijgen
2. Het systeem geeft de premie weer voor de klant van de gekozen verzekeraar

Pre-conditions: De klant is ingelogd op de webapplicatie

Post-conditions: De klant heeft de premie van een bepaalde verzekeraar verkregen

Alternatieve flow: * (1). De gebruiker selecteert geen verzekeraar

- a. Het systeem geeft een melding van het probleem
- b. ga naar (1)

Use case: Opvragen premie van een subvloot door klant

Primaire actor: De klant (een bedrijf)

Objectief: De klant moet in staat zijn de premie van een subvloot op te vragen

Normale flow:

1. De gebruiker selecteert een vloot
2. Het systeem toont informatie over de vloot
3. De gebruiker selecteert een subvloot van de vloot
4. Het systeem toont informatie over de subvloot
5. De gebruiker vraagt de premie op van de subvloot
6. Het systeem geeft de premie weer van de subvloot

Pre-conditions: De klant is ingelogd op de webapplicatie

Post-conditions: De klant kent de premie van een bepaalde subvloot

Alternatieve flow: * (5). Niet alle voertuigen hebben een premie waardoor de subvloot-premie niet weergegeven kan worden

- a. Het systeem laat weten welke voertuigen nog geen premie hebben en geeft een tussenpremie weer

- (5). Er zijn nog geen voertuigen met een premie waardoor de subvloot-premie niet weergegeven kan worden

a. Het systeem laat weten dat er alle voertuigen nog geen premie hebben

Exceptionele flow: * (1). De gebruiker heeft geen vloten

a. Het systeem geeft een melding dat er eerst vloten toegekend moeten worden

- (3). De vloot heeft geen subvloten

a. Het systeem geeft een melding dat er eerst subvloten toegekend moeten worden aan de vloot

Use case: Opvragen premie van een vloot door klant

Primaire actor: De klant (een bedrijf)

Objectief: De klant moet in staat zijn de premie van een vloot op te vragen

Normale flow:

1. De gebruiker selecteert een vloot
2. Het systeem toont informatie over de vloot
3. De gebruiker vraagt de premie op van de vloot
4. Het systeem geeft de premie weer van de vloot

Pre-condities: De klant is ingelogd op de webapplicatie

Post-condities: De klant kent de premie van een bepaalde vloot

Alternatieve flow: * (3). Niet alle voertuigen hebben een premie waardoor de vloot-premie niet weergegeven kan worden

a. Het systeem laat weten welke voertuigen nog geen premie hebben en geeft een tussenpremie weer

- (3). Er zijn nog geen voertuigen met een premie waardoor de vloot-premie niet weergegeven kan worden

a. Het systeem laat weten dat er alle voertuigen nog geen premie hebben

Exceptionele flow: * (1). De gebruiker heeft geen vloten

a. Het systeem geeft een melding dat er eerst vloten toegekend moeten worden

Use case: Opvragen premie van een voertuig door klant

Primaire actor: De klant (een bedrijf)

Objectief: De klant moet in staat zijn de premie van een voertuig op te vragen

Normale flow:

1. De gebruiker selecteert een vloot
2. Het systeem toont informatie over de vloot
3. De gebruiker selecteert een subvloot van de vloot
4. Het systeem toont informatie over de subvloot
5. De gebruiker selecteert een voertuig van de subvloot
6. Het systeem toont informatie over dit voertuig
7. De gebruiker vraagt de premie op van dit voertuig
8. Het systeem geeft de premie weer van het voertuig

Pre-condities: De klant is ingelogd op de webapplicatie

Post-condities: De klant kent de premie van een bepaald voertuig

Alternatieve flow: * (7). Het voertuig heeft geen premie waardoor de deze niet weergegeven kan worden

- a. Het systeem laat weten dat het voertuig nog geen premie heeft

Exceptionele flow: * (1). De gebruiker heeft geen vloten

a. Het systeem geeft een melding dat er eerst vloten toegekend moeten worden

- (3). De vloot heeft geen subvloten

a. Het systeem geeft een melding dat er eerst subvloten toegekend moeten worden aan de vloot

- (5). De subvloot heeft geen voertuigen

a. Het systeem geeft een melding dat er eerst voertuigen toegevoegd moeten worden aan de subvloot

Use case: Simuleren van de premiewijzigingen bij het veranderen van verzekering en/of toevoegen van voertuigen

Primaire actor: De klant (een bedrijf)

Objectief: De klant moet in staat zijn te weten wat er wijzigt bij het overschakelen naar een andere verzekering of bij het toevoegen van een voertuig

Normale flow:

1. De gebruiker geeft aan dat hij een simulatie wil starten
2. Het systeem vraagt om de huidige verzekering te wijzigen in een andere verzekering
3. De gebruiker geeft een andere verzekering op
4. Het systeem geeft de nieuwe premie weer, berekend met de gekozen verzekering in de simulatie en vraagt de gebruiker een voertuig toe te voegen, te verwijderen, te wijzigen of een andere verzekering te kiezen
5. De gebruiker voegt een voertuig toe
6. Het systeem geeft de nieuwe premie weer, berekend met de gekozen verzekering en het gewijzigde aantal en type voertuigen

Pre-condities: De klant is ingelogd op de webapplicatie

Post-condities: De klant heeft een simulatie van mogelijke premie wijzigingen gemaakt

Alternatieve flow: * (3). De gebruiker geeft geen andere verzekering op

- a. Het systeem geeft de huidige premie weer en vraagt de gebruiker een voertuig toe te voegen of te verwijderen of een andere verzekering te kiezen
 - b. Ga naar (5)
- (5). De gebruiker verwijdert een voertuig
 - a. Ga naar (6)
 - (5). De gebruiker wijzigt het type van een voertuig
 - a. Ga naar (6)
 - (5). De gebruiker wijzigt de verzekering opnieuw
 - a. Ga naar (4)

Use case: Wijzigen van wachtwoord van niet-ingelogde klant

Primaire actor: De klant (een bedrijf)

Objectief: De klant moet in staat zijn een nieuw wachtwoord aan te vragen indien hij het oude vergeten is

Normale flow:

1. De gebruiker geeft aan dat hij zijn wachtwoord vergeten is
2. Het systeem vraagt om het e-mailadres van de gebruiker

3. De gebruiker geeft zijn e-mailadres op
4. Het systeem stuurt een link naar de gebruiker om zijn wachtwoord te kunnen wijzigen
5. De gebruiker klikt op de link in de e-mail
6. Het systeem vraagt om een nieuw wachtwoord tweemaal in te geven
7. De gebruiker geeft het nieuwe wachtwoord tweemaal in
8. Het systeem verandert het wachtwoord van de gebruiker naar het nieuwe wachtwoord en stuurt een email naar de gebruiker als bevestiging dat het wachtwoord gewijzigd is

Pre-condities: De klant is niet ingelogd op de webapplicatie

Post-condities: Het wachtwoord van de klant is gewijzigd

Alternatieve flow: * (3). De gebruiker geeft een verkeerd e-mailadres in

- a. Het systeem geeft een melding dat het e-mailadres niet gevonden is
- b. Ga naar (2)
- (7). De twee ingegeven wachtwoorden komen niet overeen
 - a. Het systeem geeft een melding van het probleem
 - b. Ga naar (4)
- (7). Het ingegeven wachtwoord voldoet niet aan de minimum vereisten voor een wachtwoord
 - a. Het systeem geeft een melding van het probleem
 - b. Ga naar (4)

Use case: Aanmaken van een nieuwe medewerker of gebruiker voor bank/leasing of verzekeraar

Primaire actor: Gebruiker (administrator)

Objectief: De administrator moet een nieuwe medewerker of gebruiker voor een bank/leasing of verzekeraar kunnen toevoegen aan het systeem. (Dus geen nieuwe klant)

Normale flow

1. De administrator geeft aan dat hij een nieuwe gebruiker wil toevoegen, door naar het overzicht van de gebruikers te gaan en van daar naar de pagina voor een nieuwe gebruiker te gaan.
2. De administrator vult de gegevens voor deze gebruiker in.

3. De administrator selecteert de juiste permissies voor de nieuwe medewerker (welke klanten de medewerker beheert, of koppelt de gebruiker aan een verzekeraar of bank/leasingmaatschappij).
4. De administrator bevestigt de gegevens.
5. De administrator wordt teruggestuurd naar het overzicht van de gebruikers en krijgt een melding dat de gebruiker is toegevoegd.
6. De nieuwe gebruiker krijgt een e-mail toegestuurd met een inloglink.

Pre-condities

De gebruiker is ingelogd op de webapplicatie en heeft de nodige rechten; d.w.z. dat gebruiker de administratorrol en bijhorende permissies heeft. Indien een gebruiker voor een verzekeraar of bank/leasing toegevoegd moet worden, dient er ook minstens 1 verzekeraar of bank/leasing in het systeem te zijn.

Post-condities

Er is een nieuwe gebruiker aangemaakt.

Alternatieve flow

- (3). De gegevens zijn fout.
 1. Het systeem toont de foute gegevens, met een melding.
 2. Ga naar (2)

Use case: Aanmaken van een nieuwe gebruiker voor een klant

Primaire actor: Gebruiker (medewerker)

Objectief: De medewerker moet een nieuwe gebruiker voor een klant kunnen toevoegen aan het systeem.

Normale flow

1. De medewerker gaat naar het overzicht van zijn klanten (transportbedrijven).
2. De medewerker selecteert het bedrijf waarvoor hij een gebruiker wil toevoegen.
3. De medewerker vult de basisgegevens voor deze gebruiker in.
4. De medewerker bevestigt de gegevens.
5. De medewerker wordt naar het profiel van de nieuwe klant gestuurd.
6. De nieuwe gebruiker krijgt een e-mail toegestuurd met een inloglink.

Pre-condities

De gebruiker is ingelogd op de webapplicatie en heeft de nodige rechten; d.w.z. dat gebruiker de medewerkerrol en bijhorende permissies heeft. De medewerker heeft minstens 1 klant.

Post-condities

Er is een nieuwe gebruiker voor een klant aangemaakt.

Alternatieve flow

- (4). De gegevens zijn fout.
 1. Het systeem toont de foute gegevens, met een melding.
 2. Ga naar (2)

Use case: Verwijderen van een gebruiker

Primaire actor: Gebruiker (administrator of medewerker)

Objectief: De gebruiker moet een andere gebruiker kunnen archiveren in het systeem.

Normale flow:

1. De gebruiker gaat naar het profiel van de gebruiker.
2. De gebruiker geeft aan dat hij de gebruiker wil verwijderen.
3. De gebruiker bevestigt zijn keuze.
4. De gebruiker wordt teruggestuurd naar het overzicht van de medewerkers en krijgt een melding dat de medewerker verwijderd is.

Pre-condities

De gebruiker is ingelogd op de webapplicatie en heeft de nodige rechten; d.w.z. dat gebruiker de administratorrol of medewerkerrol heeft. Ook moet er minstens 1 andere gebruiker in het systeem zijn.

Post-condities

De gekozen gebruiker is gearhiveerd.

Alternatieve flow

- (2). De geselecteerde gebruiker kan niet verwijderd worden. Dit kan zijn omdat de te verwijderen gebruiker nog actieve dingen heeft (zoals lopende verzekeringen) of omdat de gebruiker niet de rechten heeft om de gebruiker te kunnen verwijderen (zoals een medewerker die een leasingmaatschappij probeert te verwijderen).

1. Het systeem gaat terug naar de overzichtspagina en toont een foutmelding met uitleg waarom de gebruiker niet verwijderd kon worden.
2. Ga naar (2).

Use case: Wijzigen van wachtwoord van niet-ingelogde klant

Primaire actor: De klant (een bedrijf)

Objectief: De klant moet in staat zijn een nieuw wachtwoord aan te vragen indien hij het oude vergeten is

Normale flow:

1. De gebruiker geeft aan dat hij zijn wachtwoord vergeten is
2. Het systeem vraagt om het e-mailadres van de gebruiker
3. De gebruiker geeft zijn e-mailadres op
4. Het systeem stuurt een link naar de gebruiker om zijn wachtwoord te kunnen wijzigen
5. De gebruiker klikt op de link in de e-mail
6. Het systeem vraagt om een nieuw wachtwoord tweemaal in te geven
7. De gebruiker geeft het nieuwe wachtwoord tweemaal in
8. Het systeem verandert het wachtwoord van de gebruiker naar het nieuwe wachtwoord en stuurt een email naar de gebruiker als bevestiging dat het wachtwoord gewijzigd is

Pre-condities

De klant is niet ingelogd op de webapplicatie

Post-condities

Het wachtwoord van de klant is gewijzigd

Alternatieve flow:

- (3). De gebruiker geeft een verkeerd e-mailadres in
 1. Het systeem geeft een melding dat het e-mailadres niet gevonden is
 2. Ga naar (2)
- (7). De twee ingegeven wachtwoorden komen niet overeen
 1. Het systeem geeft een melding van het probleem
 2. Ga naar (4)

- (7). Het ingegeven wachtwoord voldoet niet aan de minimum vereisten voor een wachtwoord
 1. Het systeem geeft een melding van het probleem
 2. Ga naar (4)

Use case: Een gebruiker wijzigt zijn eigen e-mail

Primaire actor: Gebruiker

Objectief: De gebruiker moet zijn eigen e-mail kunnen wijzigen.

Normale flow

1. De gebruiker gaat naar zijn eigen profielpagina.
2. De gebruiker geeft aan zijn inloggegevens te willen wijzigen.
3. De gebruiker vult zijn nieuwe e-mail in.
4. De gebruiker bevestigt de wijzigingen.
5. Het systeem stuurt de gebruiker een e-mail ter bevestiging van de wijzigingen naar het nieuwe e-mailadres.
6. Het systeem stuurt een e-mail naar het oude e-mailadres ter informatie.
7. De gebruiker opent de e-mail en klikt op bevestigen.
8. De gebruiker ziet een bevestiging van de wijzigingen.

Pre-conditions

De gebruiker is ingelogd op de webapplicatie.

Post-conditions

De gebruiker zijn gegevens zijn aangepast.

Alternatieve flow

- De gebruiker annuleert de wijzigingen.
 1. De gebruiker wordt teruggestuurd naar zijn profielpagina.
- De e-mail kon niet verstuurd worden omdat het opgeven e-mailadres fout is.
 1. De wijzigingen worden verworpen; er zal niets veranderen.
- De gebruiker negeert de bevestigingse-mail of klikt niet op de bevestigingslink binnen een bepaalde periode (zoals 24 uur, 1 week, ...).
 1. De wijzigingen worden verworpen; er zal niets veranderen.

Use case: Gegevens wijzigen van een medewerker

Primaire actor: Gebruiker (administrator)

Objectief: De administrator moet een medewerker zijn gegevens kunnen wijzigen.

Normale flow:

1. De administrator gaat naar het overzicht van de medewerkers en gaat vandaar naar het profiel van de medewerker, of komt op een andere manier bij het profiel terecht.
2. De administrator geeft aan dat hij deze medewerker zijn gegevens wil wijzigen.
3. De administrator wijzigt de gewenste gegevens.
4. De administrator bevestigt de wijzigingen.
5. De administrator krijgt het nieuwe profiel van de gebruiker te zien.

Pre-condities

De gebruiker is ingelogd op de webapplicatie en heeft de nodige rechten; d.w.z. dat gebruiker de administratorrol en bijhorende permissies heeft.

Post-condities

De gekozen gebruiker zijn gegevens zijn aangepast.

Alternatieve flow

- (3) De administrator heeft foute gegevens ingegeven.
 1. Het systeem toont een foutmelding, met de mogelijkheid om de gegevens aan te passen.
 2. Ga naar (3)
- (4) De administrator annuleert zijn wijzigen.
 1. De administrator wordt teruggestuurd naar het (ongewijzigde) profiel van de gebruiker.

Use case: Gegevens wijzigen van een gebruiker van een klant

Primaire actor: Gebruiker (medewerker)

Objectief: De medewerker moet de gegevens van de gebruiker voor een klant kunnen wijzigen.

Normale flow:

1. Een medewerker wordt op de hoogte gebracht van een wijziging in de gegevens van een gebruiker.
2. De medewerker gaat naar het profiel van de gebruiker in kwestie.
3. De medewerker geeft aan dat hij deze gebruiker zijn gegevens wil wijzigen.
4. De medewerker wijzigt de gewenste gegevens.
5. De medewerker bevestigt de wijzigingen.
6. De medewerker krijgt het nieuwe profiel van de gebruiker te zien.

Pre-condities

De gebruiker is ingelogd op de webapplicatie en heeft de nodige rechten; d.w.z. dat gebruiker de medewerkerrol en bijhorende permissies heeft. De medewerker moet minstens 1 klant hebben.

Post-condities

De gekozen klant zijn gegevens zijn aangepast.

Alternatieve flow

- (4) De medewerker heeft foute gegevens ingegeven.
 1. Het systeem toont een foutmelding, met de mogelijkheid om de gegevens aan te passen.
 2. Ga naar (3)
- (5) De medewerker annuleert zijn wijzigen.
 1. De medewerker wordt teruggestuurd naar het (ongewijzigde) profiel van de klant.
- (3) De medewerker wilt gegevens wijzigen van een gebruiker die bij een klant hoort waarvoor de medewerker geen machtigingen heeft.
 1. De medewerker wordt teruggestuurd naar het (ongewijzigde) profiel van de klant.

Use case: Wijzigen van de rechten van een gebruiker

Primaire actor: Gebruiker (medewerker of administrator)

Objectief: De gebruiker moet een andere gebruiker zijn rechten kunnen aanpassen.

Normale flow:

1. De gebruiker gaat naar een andere gebruiker zijn profiel.
2. De gebruiker geeft aan dat hij deze gebruiker zijn rechten wil wijzigen.

3. De gebruiker wijzigt de gewenste rechten.
4. De gebruiker bevestigt de wijzigingen.
5. De gebruiker wordt doorgestuurd naar het profiel van de gebruiker, met een melding dat de rechten aangepast zijn.

Pre-conditions

De gebruiker is ingelogd op de webapplicatie en heeft de nodige rechten. Het is dus een gebruiker met de rol administrator of medewerker.

Post-conditions

De rechten van de gekozen gebruiker zijn gewijzigd.

Alternatieve flow

- (2) De gebruiker probeert de rechten van een gebruiker te wijzigen waarvoor hij het recht niet heeft. Dit gebeurt bijvoorbeeld als een medewerker de rechten van een andere medewerker probeert te wijzigen, aangezien dit enkel door de administrator kan gebeuren.
 1. Het systeem toont een foutmelding.
 2. Ga naar (1).
- (3) De gebruiker probeert rechten toe te kennen waartoe hij zelf geen rechten heeft. Dit gebeurt bijvoorbeeld als een medewerker rechten voor het beheer van een vloot aan een klant probeert toe te kennen.
 1. Het systeem toont een foutmelding, met de mogelijkheid om de gekozen rechten aan te passen.
 2. Ga naar (3)
- (4) De gebruiker annuleert zijn wijzigen.
 1. De gebruiker wordt teruggestuurd naar het (ongewijzigde) profiel van de gebruiker

Use case: Wijzigen van wachtwoord van ingelogde klant

Primaire actor: De klant (een bedrijf)

Objectief: De klant moet in staat zijn een nieuw wachtwoord aan te vragen

Normale flow:

1. De gebruiker geeft aan dat hij zijn wachtwoord wil wijzigen
2. Het systeem vraagt om het oude (huidige) wachtwoord in te voeren
3. De gebruiker geeft het oude (huidige) wachtwoord in

4. Het systeem vraagt om het nieuwe wachtwoord tweemaal in te geven
5. De gebruiker geeft het nieuwe wachtwoord tweemaal in
6. Het systeem verandert het wachtwoord van de gebruiker naar het nieuwe wachtwoord en stuurt een email naar de gebruiker als bevestiging dat het wachtwoord gewijzigd is

Pre-condities: De klant is ingelogd op de webapplicatie

Post-condities: Het wachtwoord van de klant is gewijzigd

Alternatieve flow: * (3). De gebruiker geeft het verkeerde wachtwoord in

- a. Het systeem geeft een melding van het probleem
- b. Ga naar (2)
- (3). De gebruiker geeft het verkeerde wachtwoord herhaaldelijk verkeerd in
 - a. Het systeem geeft een melding van het probleem
 - b. De gebruiker dient een captcha op te lossen
 - c. Ga naar (2)
- (5). De twee ingegeven wachtwoorden komen niet overeen
 - a. Het systeem geeft een melding van het probleem
 - b. Ga naar (4)
- (5). Het ingegeven wachtwoord voldoet niet aan de minimum vereisten voor een wachtwoord
 - a. Het systeem geeft een melding van het probleem
 - b. Ga naar (4)

Voertuig

Use case: Voertuigen van een subvloot ophalen

Primaire actor: Een gebruiker

Objectief: De gebruiker wil de subvloten van een klant ophalen

Normale flow:

1. De gebruiker gaat naar de pagina van de klant
2. De gebruiker kiest de juiste vloot
3. (optioneel) de gebruiker geeft eventuele filters en/of sorteervormen mee

4. De gebruiker bevestigt de zoekopdracht
5. De gebruiker krijgt de lijst te zien

Pre-condities: - De gebruiker is ingelogd op de webapplicatie en heeft de nodige rechten - De vloot waarvan de gebruiker subvloten wil zien bestaat - De klant heeft een profiel in het systeem

Post-condities: - Zowel de klant als de verzekeringsmaatschappij kunnen deze verandering bekijken

Use case: Voertuig aan een subvloot toevoegen

Primaire actor: Productie- of schadebeheerder

Objectief: De productie- of schadebeheerder moet een voertuig toevoegen aan een subvloot van een klant

Normale flow:

1. De klant neemt contact op met solvas en geeft de nieuwe voertuigen door
2. De medewerker van Solvas zoekt de klant op in de lijst van klanten
3. De medewerker selecteert de juiste klant
4. De medewerker zoekt de subvloot op in de lijst van subvloten
5. De medewerker selecteert de juiste subvloot
6. De medewerker geeft aan dat hij een voertuig wil toevoegen
7. De medewerker geeft de gegevens van het voertuig door (uniek chassisnummer,...)
8. Het systeem vraagt om bevestiging
9. De medewerker bevestigt de toevoeging en slaat deze op in de databank

Pre-condities: - De medewerker is ingelogd op de webapplicatie en heeft de nodige rechten - De klant heeft een profiel in het systeem

Post-condities: Het voertuig werd toegevoegd

Alternatieve flow

- (4). De klant heeft nog geen vloten
 - a. Het systeem toont een foutmelding
 - b. Er wordt een nieuwe vloot gemaakt na bespreking met de klant
 - c. Ga naar 4
- (4). De klant heeft geen subvloot voor het type voertuig

- a. Het systeem toont een foutmelding
- b. Er wordt een nieuwe subvloot aangemaakt na bespreking met de klant
- c. Ga naar 4
- (7). Het chassinummer werd al gebruikt bij een andere subvloot
 - a. Het systeem toont een foutmelding
 - b. De medewerker neemt eventueel contact op met solvas
- (7). Gegevens staan in een fout formaat
 - a. Het systeem toont een foutmelding met mogelijkheid om gegevens aan te passen
 - b. Ga naar 7
- (8). De medewerker annuleert de verwijdering
 - a. De medewerker bedenkt zich en drukt op annuleren

Use case: Voertuig verwijderen van een subvloot

Primaire actor: Productie- of schadebeheerder

Objectief: De productie- of schadebeheerder moet een voertuig verwijderen van een subvloot van een klant

Normale flow:

1. De klant neemt contact op met solvas en vraagt de verwijdering van het voertuig aan
2. De medewerker van Solvas zoekt de klant op in de lijst van klanten
3. De medewerker selecteert de juiste klant
4. De medewerker zoekt het voertuig in de lijst van voertuigen
5. De medewerker selecteert het juiste voertuig
6. De medewerker geeft aan dat hij het voertuig wil verwijderen
7. Het systeem vraagt om bevestiging
8. De medewerker geeft bevestiging

Pre-condities: - De medewerker is ingelogd op de webapplicatie en heeft de nodige rechten - De klant heeft een profiel in het system - Het voertuig staat in de subvloot

Post-condities: - Het voertuig werd verwijderd uit de subvloot - Het voertuig bestaat nog steeds in het systeem

Alternatieve flow: * (7). De medewerker annuleert de verwijdering

- a. De medewerker bedenkt zich en drukt op annuleren
- (6). Het voertuig heeft nog een active verzekering
- a. Het systeem toont een foutmelding die het probleem beschrijft
- b. De active verzekering word stop gezet
- c. Ga naar 7

Use case: Voertuig wijzigen

Primaire actor: Productie- of schadebeheerder

Objectief: De productie- of schadebeheerder moet een voertuig verwijderen van een subvloot van een klant

Normale flow:

1. De klant neemt contact op met solvas en geeft de wijzeging van het voertuig door
2. De medewerker van Solvas zoekt de klant op in de lijst van klanten
3. De medewerker selecteert de juiste klant
4. De medewerker zoekt het voertuig in de lijst van voertuigen
5. De medewerker selecteert het juiste voertuig
6. De medewerker geeft aan dat hij het voertuig wil wijzigen
7. De medewerker geeft de wijzigingen door
8. Het systeem vraagt om bevestiging
9. De medewerker geeft bevestiging

Pre-condities: - De medewerker is ingelogd op de webapplicatie en heeft de nodige rechten - De klant heeft een profiel in het systeem - Het voertuig staat in de subvloot

Post-condities: - Het voertuig werd aangepast

Alternatieve flow: * (7). Gegevens staan in een fout formaat

- a. Het systeem toont een foutmelding met mogelijkheid om gegevens aan te passen
- b. Ga naar 7
- (9). De medewerker annuleert de wijziging
- a. De medewerker bedenkt zich en annuleerd de wijzeging

Subvloot

Use case: Voertuigen van een subvloot van een klant ophijsten

Primaire actor: Een gebruiker

Objectief: De gebruiker wil de voertuigen van een subvloot van een klant ophijsten

Normale flow:

1. De gebruiker gaat naar de pagina van de klant
2. De gebruiker kiest de juiste vloot en subvloot
3. De gebruiker (optioneel) de gebruiker geeft eventuele filters en/of sorteer-
vormen mee
4. De gebruiker bevestigt de zoekopdracht
5. De gebruiker krijgt de lijst te zien

Pre-condities: - De gebruiker is ingelogd op de webapplicatie en heeft de
nodige rechten - De subvloot bestaat - De klant heeft een profiel in het systeem

Use case: Subvloot van een klant toevoegen

Primaire actor: Productie- of schadebeheerder

Objectief: De productie- of schadebeheerder moet een subvloot van een klant
toevoegen aan de databank

Normale flow:

1. De medewerker van Solvas zoekt de klant op in de lijst van klanten
2. De medewerker selecteert de juiste klant
3. De medewerker geeft aan dat hij een subvloot wil toevoegen aan de klant
4. De medewerker geeft het voertuigtype door, horend bij de subvloot
5. De medewerker bevestigt de toevoeging en slaat deze op in de databank

Pre-condities: - De gebruiker is ingelogd op de webapplicatie en heeft de
nodige rechten - De klant staat in de databank

Post-condities: De subvloot is klaar om er voertuigen aan toe te voegen.

Use case: Subvloot van een klant wijzigen

Primaire actor: Productie- of schadebeheerder

Objectief: De productie- of schadebeheerder moet de subvloot van een klant verwijderen. Hierdoor is deze alleen zichtbaar voor de klant als hij de historiek nagaat.

Normale flow:

1. (De klant contacteert Solvas voor de verwijdering uit te voeren)
2. De medewerker van Solvas zoekt de klant op in de lijst van klanten
3. De medewerker selecteert de juiste klant
4. De medewerker selecteert de subvloot
5. De medewerker verwijdert de subvloot
6. Het systeem vraagt om bevestiging
7. De medewerker bevestigt de verwijdering

Pre-condities: - De gebruiker is ingelogd op de webapplicatie en heeft de nodige rechten - De subvloot bestaat

Post-condities: - De klant kan de subvloot enkel maar bekijken als hij de historiek nagaat

Use case: Subvloot van een klant wijzigen

Primaire actor: Productie- of schadebeheerder

Objectief: De productie- of schadebeheerder moet de subvloot van een klant wijzigen. Het wijzigen gaat dus over de verzekering bij de subvloot.

Normale flow:

0. (De klant contacteert Solvas om de verzekering van een subvloot te wijzigen)
1. De medewerker van Solvas zoekt de klant op in de lijst van klanten
2. De medewerker selecteert de juiste klant
3. De medewerker selecteert de subvloot
4. De medewerker verandert de verzekering, met begin- en einddatums
5. Het systeem vraagt om bevestiging
6. De medewerker bevestigt de wijziging en slaat deze op in de databank

Pre-condities: - De gebruiker is ingelogd op de webapplicatie en heeft de nodige rechten - De subvloot bestaat - De klant heeft een profiel

Post-condities: - Zowel de klant als de verzekeringsmaatschappij kunnen deze verandering bekijken

Alternatieve flow: * (4). De begin- en/of einddatum zijn niet correct

a. Het systeem toont een foutmelding, met de mogelijkheid om de data aan te passen

b. Ga naar (5)

• (6). De medewerker wilt de aanpassing annuleren

a. De medewerker drukt op annuleren

b. De medewerker krijgt de subvloot in normale staat te zien

Verzekeringsmaatschappij

Use case: Facturen van een verzekeringsmaatschappij opvragen

Primaire actor: Medewerker verzekeringsmaatschappij (voortaan gewoonweg medewerker genoemd)

Objectief: De verzekeringsmaatschappij moet een overzicht kunnen krijgen met alle facturen van alle aangesloten makelaars.

Normale flow:

1. De medewerker geeft aan dat hij de facturen wil opvragen.
2. Er wordt gevraagd naar een start en einddatum waartussen hij wil zoeken (dit is verplicht). Hierbovenop kunnen optionele parameters zoals een bepaalde makelaar, betaalstatus (betaald of nog niet betaald) of bedraginterval kunnen meegegeven worden.
3. De medewerker bevestigt de parameters en geeft deze aan het systeem door.
4. Het systeem stelt intern deze lijst op en toont deze aan de medewerker.
5. De zoekparameters kunnen veranderd worden waardoor men terugvalt naar stap 2 of men kan deze lijst printen.

Pre-condities: De medewerker is ingelogd op de webapplicatie en is verbonden met een verzekeringsmaatschappij.

Post-condities: De medewerker ontvangt een lijst van de facturen waarvoor hij geïnteresseerd is.

Alternatieve flow:

- (2). De gegevens zijn fout vb (einddatum < startdatum,..) a. Het systeem toont de foutmelding.
b. Ga naar stap 2 en probeer opnieuw.
- (5). De printer werd niet gevonden of er vond een andere IO-fout plaats.
a. Het systeem toont de foutmelding afkomstig van de printer.
b. Ga naar stap 5 en probeer opnieuw.

Exceptionele flow:

- (1). De medewerker heeft geen rechten om de facturen te bekijken. Het systeem toont een foutmelding, met de mogelijkheid om rechten aan te vragen aan een overste van de verzekeringsmaatschappij.
- (4). De medewerker heeft geen rechten om de facturen te bekijken. Het systeem toont een foutmelding, met de mogelijkheid om rechten aan te vragen aan een overste van de verzekeringsmaatschappij.

Use case: Nog niet betaalde facturen van een verzekeringsmaatschappij bekijken

Primaire actor: Medewerker verzekeringsmaatschappij (voortaan gewoonweg medewerker genoemd)

Objectief: De verzekeringsmaatschappij moet kunnen zien welke facturen nog niet betaald geweest zijn door de aangesloten makelaars.

Normale flow:

1. <<include “facturen opvragen”>> De medewerker geeft aan het systeem de waarde “niet betaald” mee voor de zoekparameter “betaalstatus”.
2. Het systeem toont de gewenste lijst aan de medewerker.

Pre-conditions: De medewerker is ingelogd op de webapplicatie en is aangesloten als werknemer bij een verzekeringsmaatschappij.

Post-conditions: De medewerker ziet een lijst van alle niet-betaalde facturen.

Exceptionele flow:

- (1). De medewerker heeft geen rechten om de facturen te bekijken. Het systeem toont een foutmelding, met de mogelijkheid om rechten aan te vragen aan een overste van de verzekeringsmaatschappij.

Extra

Use case: Importeren van Data

Primaire actor: Gebruiker

Objectief: De gebruiker moet data via een csv (of excel) data in bulk kunnen importeren in het database systeem.

Normale flow:

1. De gebruiker geeft aan dat hij een import wil starten
2. De gebruiker kiest zijn bestand om te importeren
3. Het systeem toont welke wijzigingen aangebracht zullen worden aan het databasesysteem
4. De gebruiker bevestigt de wijzigingen
5. De gebruiker wordt doorgestuurd naar een bevestigingspagina, en zal later een notificatie krijgen dat de wijzigingen succesvol zijn doorgevoerd

Pre-condities: De gebruiker is ingelogd op de webapplicatie en heeft de nodige rechten

Post-condities: De gebruiker heeft zijn gegevens geïmporteerd

Alternatieve flow: * (3). Het bestand bevat foute gegevens

- a. Het systeem toont de foute gegevens, met een melding
- b. Ga naar (2)
- (4). De gebruiker aanvaardt de wijzigingen niet
- a. Ga naar (1)

Use case: Exporteren van Data

Primaire actor: Gebruiker

Objectief: De gebruiker moet (een deel van) het databasesysteem kunnen exporteren naar een csv

Normale flow:

1. De gebruiker geeft aan dat hij een export wil starten
2. De gebruiker kiest de gegevens die hij wil exporteren
3. De gebruiker kiest het formaat waarin hij de data wil exporteren
4. De gebruiker bevestigt zijn export

5. Het systeem stuurt de gebruiker door naar een bevestigingspagina
6. Het systeem stuurt een mail met de export naar de gebruiker

Pre-condities: De gebruiker is ingelogd op de webapplicatie en heeft de nodige rechten

Post-condities: De gebruiker heeft zijn data geëxporteerd

Use case: afsluiten van een verzekering

Primaire actor: De beheerder

Objectief: Er wordt een verzekering afgesloten tussen de klant en een verzekeringsmaatschappij over een voertuig.

Normale flow:

1. De beheerder selecteert het account van de klant.
2. De beheerder selecteert het voertuig waarvoor de klant een verzekering wilt hebben door de juiste vloot, subvloot te kiezen.
3. De klant geeft aan welk type verzekering hij wilt voor het voertuig.
4. De klant kan uit een lijst kiezen welke verzekering hij/zij wilt.
5. De beheerder selecteert deze verzekering en slaat de upgedate gegevens op

Pre-condities: De klant heeft een profiel bij solvas en heeft minstens 1 voertuig in zijn vloot alsook is de beheerder ingelogd in het systeem.

Post-condities: Het voertuig heeft een nieuwe verzekering

Alternatieve flow: * (1). De beheerder heeft de rechten niet om het account van de klant te beheren a. De beheerder vraagt deze rechten aan aan de administrator

- b. Ga naar (2)
- (3). Het gekozen voertuig heeft al een verzekering voor op het voertuig voor het gekozen type
- a. Het systeem geeft een melding van het probleem
- b. Huidige verzekering stopzetten
- c. Ga naar (4)