

# SIE-lab boekingsysteem

Functioneel ontwerp

Bij Jordy Koppers, Thim Eickhof, Gideon de Bruijn

# Inhoudsopgave

<b>Inleiding</b>	<b>3</b>
<b>Lifecycle</b>	<b>4</b>
<b>Actoren</b>	<b>5</b>
<b>Use cases</b>	<b>6</b>
<b>Volledigheid Analyse</b>	<b>8</b>
<b>Sitemap</b>	<b>9</b>
<b>Use case 1: Ruimte boeken</b>	<b>11</b>
<b>Use case 2: Boeking beoordelen</b>	<b>13</b>
<b>Use case 3: Boeking bekijken</b>	<b>15</b>
<b>Use case 4: Boeking verwijderen</b>	<b>16</b>
<b>Use case 5: QR-Code scannen</b>	<b>17</b>
<b>Use case 6: Account verifiëren</b>	<b>18</b>
<b>Wireframes</b>	<b>20</b>

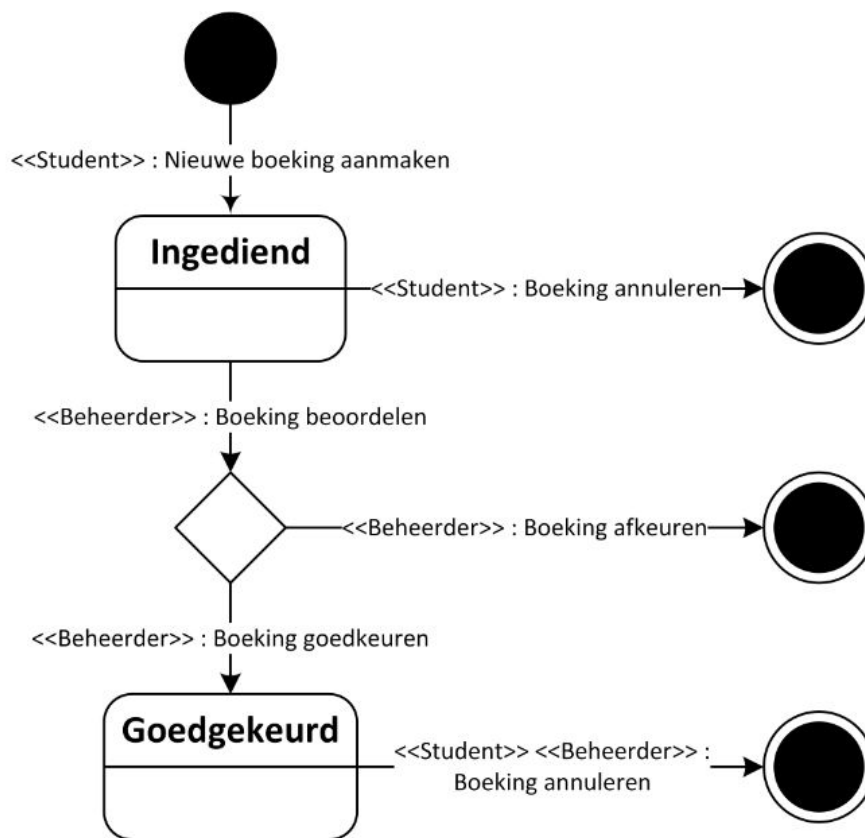
# Inleiding

Voor het vak Front End Development moet er een webapp gebouwd worden om het gebruik van het Software Development Lab te beheren. De app wordt gebruikt om de ruimte te reserveren en moet ervoor zorgen dat alleen studenten die de ruimte daadwerkelijk hebben gereserveerd toegang krijgen tot het Lab. Dit document, het Functioneel Ontwerp, is de basis van de webapp.

Het FO beschrijft de functionaliteiten die de app moet bevatten. Eerst wordt het proces beschreven met een toestandsdiagram. Daarna worden de functionaliteiten verder uitgewerkt door middel van een Use Case Diagram met uitgewerkte Use Case Templates. Ook is er een volledighedsanalyse, sitemap en wireframes voor elke pagina die gerealiseerd gaat worden.

# Lifecycle

Het kern-object van de applicatie is de boeking van het Lab. In de applicatie maakt een student een nieuwe boeking aan, deze boeking moet door de beheerder beoordeeld worden. Hij kan de boeking goed- of afkeuren. Als de boeking goedgekeurd is kan de student gebruik maken van het Lab. Tijdens het hele proces kan de student de boeking annuleren. Als de boeking is goedgekeurd kan de beheerder de boeking ook nog annuleren. Als we dit proces vertalen naar een toestandsdiagram ziet dat er als volgt uit.



Figuur 1: Lifecycle

# Actoren

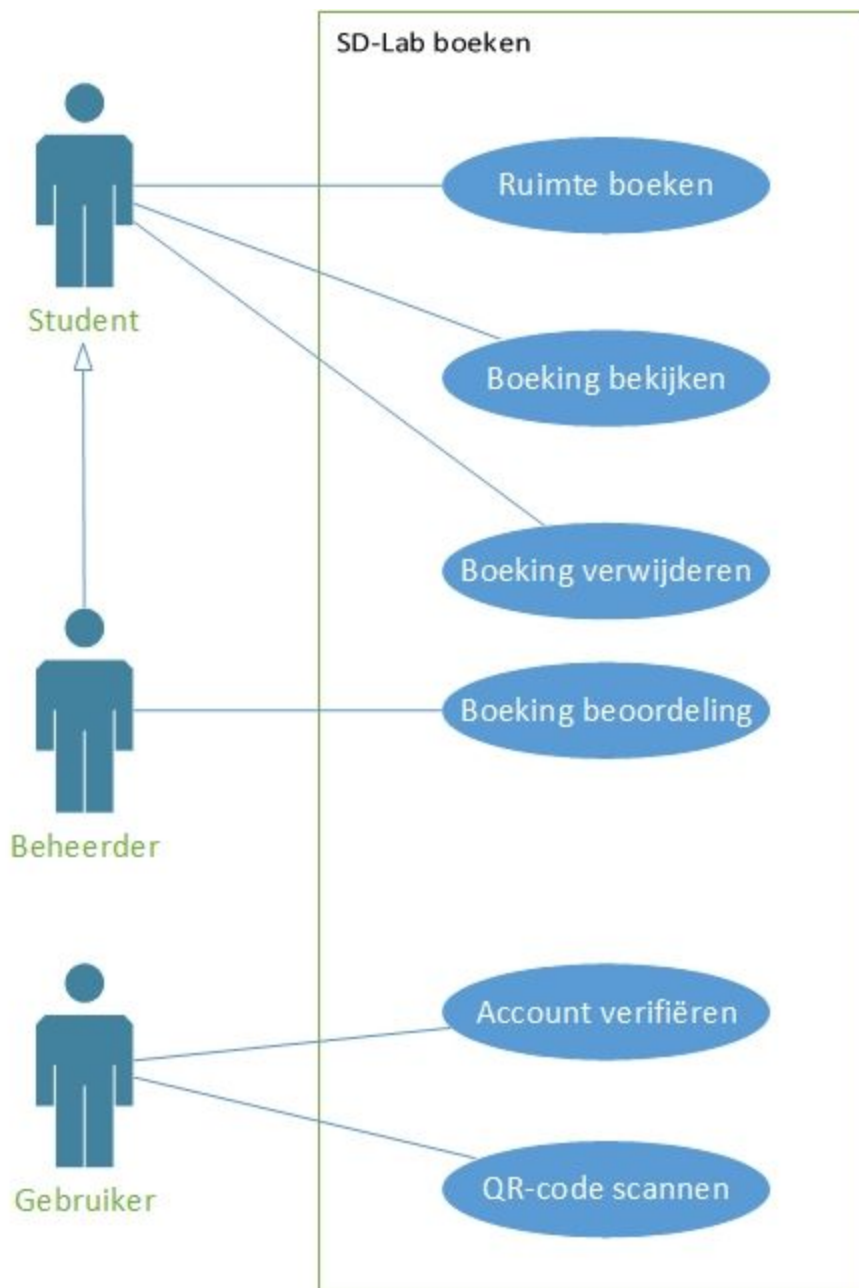
Er zijn 3 actoren bedacht om het systeem alle functionaliteiten te laten ondersteunen die verwacht worden.

Project:	SIE-lab boekingssysteem
Rolnaam:	<b>Gebruiker</b>
Rolbeschrijving:	De gebruiker zijn alle personen totdat ze een ander actor worden door in te loggen met een geverifieerd account. Deze rol bestaat zodat de site voor iedereen toegankelijk is en vanuit daar een andere actor kan worden.

Project:	SIE-lab boekingssysteem
Rolnaam:	<b>Student</b>
Rolbeschrijving:	De student is de ingelogde gebruiker. Om te zorgen dat niet iedereen zomaar in het systeem aanmeldingen kan doen, heeft deze actor toegang tot de use cases die leiden naar het kerndoel van dit systeem, het (aanmelden tot) boeken van het SDLab.

Project:	SIE-lab boekingssysteem
Rolnaam:	<b>Beheerder</b>
Rolbeschrijving:	De beheerder is de beheerder van het systeem en heeft toegang tot alles. Deze actor is in dit geval ook de opdrachtgever. Om het kerndoel te bereiken en een check te doen op welke studenten het SDLab gebruiken is deze actor nodig.

# Use cases



Figuur 2: Use Case diagram

In dit Use Case Diagram zijn drie Actoren en vijf Use Cases te zien.

Inloggen en Uitloggen zijn niet opgenomen als use cases want dat zijn acties die bepalen welke actor je bent.

Per use case kunnen we nu een samenvatting opstellen:

Use Case	Samenvatting
Ruimte boeken	Het systeem vraagt om een <b>datum(in)</b> , <b>begintijd(in)</b> , <b>eindtijd(in)</b> en optioneel een <b>opmerking(in)</b> . De actor verstrekt deze gegevens. Het systeem vraagt om een bevestiging en de actor bevestigt.
Boeking beoordelen	Het systeem laat per boeking <b>studentenmail(out)</b> , <b>studentennummer(out)</b> , <b>datum(out)</b> , <b>begintijd(out)</b> en <b>eindtijd(out)</b> en de optie om deze goed of af te keuren om ruimte te boekingen zien. De actor keurt een boeking goed. Het systeem verifieert de boeking. Het systeem genereert een QR-code. Het systeem stuurt de QR-code naar de studentenmail van de applicant. Het systeem zet de boeking in de google calendar.
Boeking bekijken	Het systeem toont per boeking de <b>schoolmail(out)</b> , <b>datum(out)</b> , <b>begintijd(out)</b> , <b>eindtijd(out)</b> en <b>opmerking(out)</b> voor de ingelogde user(of alle boekingen voor de beheerder).
Boeking verwijderen	Systeem laat alle <b>boekingen(out)</b> zien. Actor selecteert een boeking om te verwijderen. Systeem vraagt om bevestiging. Actor bevestigt. Systeem verwijdert de boeking uit het systeem.
QR-Code scannen	Systeem vraagt om QR-code. Actor laat de QR-code scannen. Systeem bevestigt de QR-code en opent de deur.
Account verifiëren	Het systeem vraagt om een <b>studentnummer(in)</b> , <b>voornaam(in)</b> , <b>achternaam(in)</b> , <b>studentenmail(in)</b> . De actor verstrekt deze gegevens. Het systeem vraagt om een bevestiging en de actor bevestigt. Het systeem genereert een verificatiecode en slaat deze op, het systeem stuurt een email met verificatiecode naar de studentenmail en vraagt om de verificatiecode. Actor vult deze gegevens in. Systeem verifieert het account in het systeem.

# Volledigheid Analyse

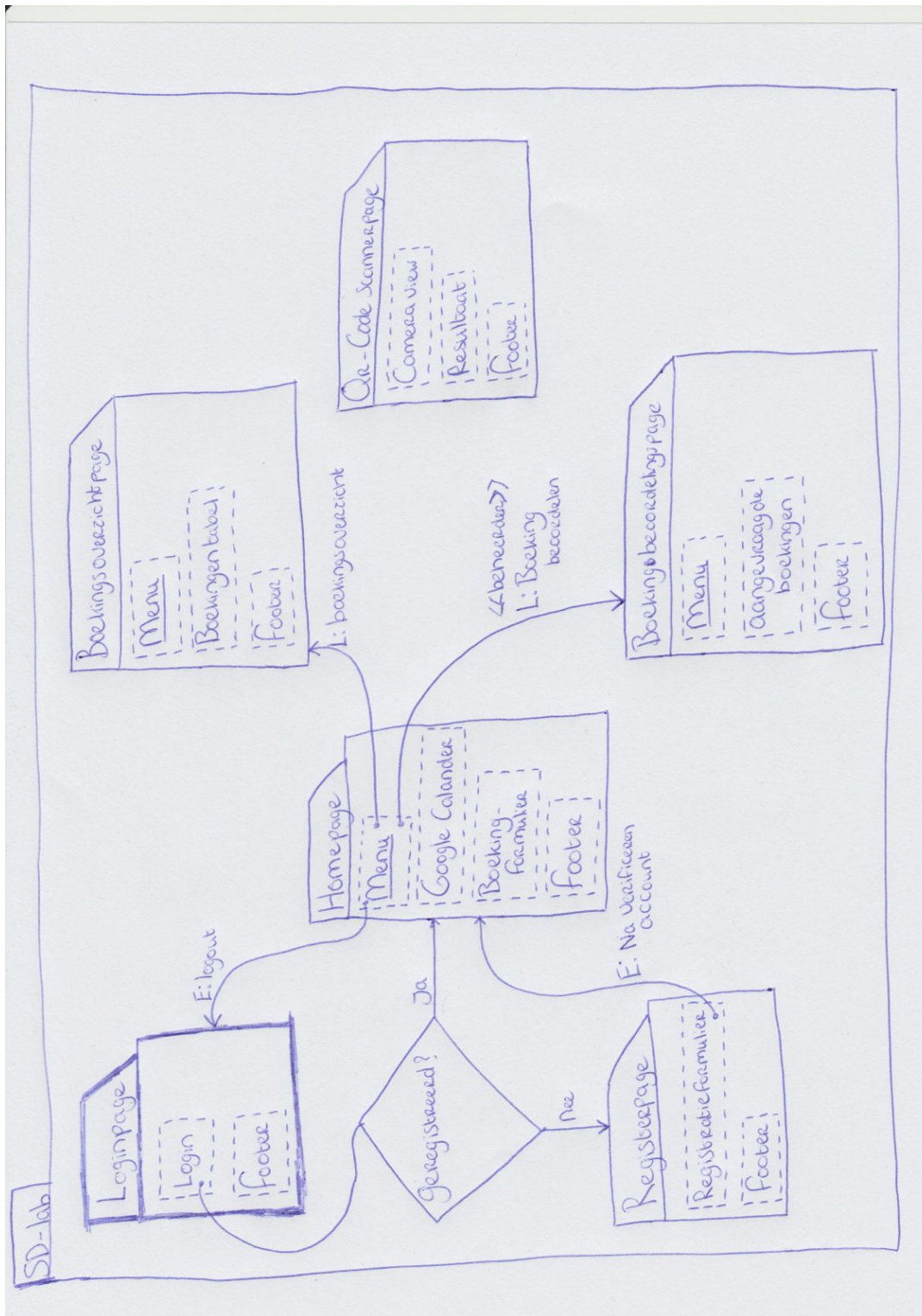
Met de bovenstaande use case samenvattingen kan er nu een volledigheidcheck gedaan worden van het systeem.

	Ruimte boeken	Boeking beoordelen	Boeking bekijken	Boeking verwijderen	QR-Code scannen	Account verifiëren	Conclusie
Datum	in	out					OK
Begintijd	in	out					OK
Eindtijd	in	out					OK
Opmerking	in	out	out				OK
Studentenmail		out				in	OK
Boeking		out	out				
Studentennummer		out				in	OK
Voornaam		out				in	OK
Achternaam		out				in	OK

In het bovenstaande tabel is te zien dat het attribuut boekingen alleen maar outputs heeft, dit komt omdat een boeking bestaat uit een datum, begintijd, eindtijd, opmerking, studentenmail, studentennummer, voornaam en achternaam. Deze attributen hebben wel allemaal inputs en daarom kan een boeking output geven.



# Sitemap



Figuur 3: Sitemap

**Beschrijving sitemap:**

De gebruiker begint bij de loginpagina, hier kan de gebruiker inloggen met een Google account. Nadat de gebruiker is ingelogd met zijn Google account wordt er in het systeem gekeken of dit Google account al gekoppeld is aan een studentenmail. Als dit niet het geval is komt de gebruiker op de registratiepagina. Hier moet de gebruiker zijn studentenmail en zijn studentnummer invoeren. Daarna krijgt de gebruiker op zijn studentenmail een verificatielink/code waarmee de gebruiker kan aantonen dat de studentenmail valide is daarna wordt de gebruiker doorgestuurd naar de homepage. Als de gebruiker na het inloggen met een Google account wel bekend is in het systeem komt de gebruiker direct op de homepage. Op de homepage ziet de gebruiker de agenda van het SD-lab, hierin is te zien wanneer het SD-lab beschikbaar is. Onder deze agenda is een invullijst te vinden waar de gebruiker het lokaal kan boeken voor een bepaalde periode. Ook kan de gebruiker zich via de navbar naar het boekingsoverzicht navigeren. Hier ziet de gebruiker al zijn boekingen, boekingen die goedgekeurd zijn hebben een downloadlink van een qr-code waarmee toegang te krijgen is tot het SD-lab in de geboekte periode. Als de gebruiker een beheerder is, is er een overzicht te zien van alle boekingen. Als beheerder is er nog een optie in de navbar, dit is de optie om een boeking te kunnen beoordelen. Als de beheerder op deze optie in de navbar klikt komt hij op een pagina waarin boekingen te zien zijn die nog niet zijn beoordeeld. De beheerder kan op deze pagina boekingen goedkeuren of afkeuren. Ten slotte is er nog een laatste pagina waarmee een QR-code gescand kan worden. Deze pagina wordt alleen gebruikt op de Raspberry Pi die in het SD-lab staat. Deze kan hiermee kijken of er een valide QR-code gescand wordt. Als dit zo is wordt het scherm groen, zoniet dan wordt het scherm rood.

# Use case 1: Ruimte boeken

<b>ID:</b> 1	<b>Naam:</b> Ruimte boeken
<b>Actor:</b> Student, Beheerder	
<b>Samenvatting:</b> Het systeem vraagt om een <b>datum(in)</b> , <b>begintijd(in)</b> , <b>eindtijd(in)</b> en optioneel een <b>opmerking(in)</b> . De actor verstrekt deze gegevens. Het systeem vraagt om een bevestiging en de actor bevestigt.	
<b>Precondities:</b> geen	
<b>Main scenario:</b> <ol style="list-style-type: none"><li>1. Het systeem vraagt om een <b>datum(in)</b>, <b>begintijd(in)</b>, <b>eindtijd(in)</b> en optioneel een <b>opmerking(in)</b>.</li><li>2. De actor verstrekt deze gegevens.</li><li>3. Het systeem zet de aanmelding in het systeem.</li></ol>	
<b>Postcondities:</b> Aanmelding om ruimte te boeken is aangemaakt.	
<b>Alternatief scenario trigger (A1):</b> In stap 2 van het main scenario. [Actor annuleert usecase.]  <b>Alternatief scenario stappen(A1):</b> <ol style="list-style-type: none"><li>1. Systeem beëindigt de usecase.</li></ol> <b>Post conditie:</b> Systeem is ongewijzigd.	
<b>Alternatief scenario trigger (A2):</b> In stap 2 van het main scenario. [Actor voert niet alle benodigde gegevens in.]  <b>Alternatief scenario stappen(A2):</b> <ol style="list-style-type: none"><li>1. Systeem toont foutmelding.<ol style="list-style-type: none"><li>1.1 Systeem begint opnieuw met stap 1 van het main scenario.</li></ol></li></ol>	

**Alternatief scenario trigger(A3):**

Na stap 3 van het main scenario  
[ruimte is al geboekt]

**Alternatief scenario stappen(A3):**

1.

1.1 Systeem toont melding dat de ruimte al geboekt is.

1.2 Systeem systeem gaat verder met stap2 van het main scenario

**Post conditie:**

Systeem is ongewijzigd.

## Use case 2: Boeking beoordelen

<b>ID:</b> 2	<b>Naam:</b> Boeking beoordelen
<b>Actor:</b> Beheerder	
<b>Samenvatting:</b> Het systeem laat per boeking <b>studentenmail(out)</b> , <b>studentennummer(out)</b> , <b>datum(out)</b> , <b>begintijd(out)</b> en <b>eindtijd(out)</b> en de optie om deze goed of af te keuren om ruimte te boekingen zien. De actor keurt een boeking goed. Het systeem verifieert de boeking. Het systeem genereert een QR-code. Het systeem stuurt de QR-code naar de studentenmail van de applicant. Het systeem zet de boeking in de google calendar.	
<b>Precondities:</b> geen.	
<b>Main scenario:</b> <ol style="list-style-type: none"><li>1. Het systeem laat per boeking <b>studentenmail(out)</b>, <b>studentennummer(out)</b>, <b>datum(out)</b>, <b>begintijd(out)</b> en <b>eindtijd(out)</b> en de optie om deze goed of af te keuren per ruimte te boekingen zien.</li><li>2. De actor keurt een boeking goed.</li><li>3. 3.1 Het systeem verifieert de boeking. 3.2 Het systeem genereert een QR-code. 3.3 Het systeem stuurt de QR-code naar de studentenmail van de applicant. 3.4 Het systeem zet de boeking in de google calendar.</li></ol>	
<b>Postcondities:</b> Ruimte is geboekt, QR-code is verstuurt en de Google calendar is geupdate.	
<b>Alternatief scenario trigger (A1):</b> In stap 2 van het main scenario. [Actor annuleert usecase.]  <b>Alternatief scenario stappen(A1):</b> <ol style="list-style-type: none"><li>1. Systeem beëindigt de usecase.</li></ol> <b>Post conditie:</b> Systeem is ongewijzigd.	
<b>Alternatief scenario trigger (A1):</b> In stap 3 van het main scenario. [Aanmelding bestaat niet meer]  <b>Alternatief scenario stappen(A2):</b>	

1. 1. Systeem toont foutmelding.
2. Systeem begint opnieuw met stap 1 van het main scenario.

**Post conditie:**

Systeem is ongewijzigd.

## Use case 3: Boeking bekijken

<b>ID:</b> 3	<b>Naam:</b> Boeking inkijken
<b>Actor:</b> Student, Beheerder	
<b>Samenvatting:</b> Het systeem toont per boeking de <b>schoolmail(out)</b> , <b>datum(out)</b> , <b>begintijd(out)</b> , <b>eindtijd(out)</b> en <b>opmerking(out)</b> voor de ingelogde user(of alle boekingen voor de beheerder).	
<b>Precondities:</b> geen.	
<b>Main scenario:</b> <ol style="list-style-type: none"><li>1. Het systeem toont per boeking de <b>schoolmail(out)</b>, <b>datum(out)</b>, <b>begintijd(out)</b>, <b>eindtijd(out)</b> en <b>opmerking(out)</b> voor de ingelogde user(of alle boekingen voor de beheerder).</li></ol>	
<b>Postcondities:</b> geen.	

## Use case 4: Boeking verwijderen

<b>ID:</b> 4	<b>Naam:</b> Boeking verwijderen
<b>Actor:</b> Student, Beheerder	
<b>Samenvatting:</b> Het systeem toont per boeking de <b>schoolmail(out)</b> , <b>datum(out)</b> , <b>begintijd(out)</b> , <b>eindtijd(out)</b> en <b>opmerking(out)</b> voor de ingelogde user(of alle boekingen voor de beheerder) met de optie om deze te verwijderen. Actor kiest de optie om een boeking te verwijderen. Systeem verwijdert de boeking uit het systeem.	
<b>Precondities:</b> geen.	
<b>Main scenario:</b> <ol style="list-style-type: none"><li>1. Het systeem toont per boeking de <b>schoolmail(out)</b>, <b>datum(out)</b>, <b>begintijd(out)</b>, <b>eindtijd(out)</b> en <b>opmerking(out)</b> voor de ingelogde user(of alle boekingen voor de beheerder) met de optie om deze te verwijderen.</li><li>2. De actor kiest de optie om een boeking te verwijderen.</li><li>3. Het systeem verwijdert de boeking uit het systeem.</li></ol>	
<b>Postcondities:</b> De geselecteerde boeking is verwijderd uit het systeem.	
<b>Alternatief scenario trigger (A1):</b> In stap 2 van het main scenario. [Actor annuleert use case.]  <b>Alternatief scenario stappen(A1):</b> <ol style="list-style-type: none"><li>1. Systeem beëindigt de use case.</li></ol> <b>Post conditie:</b> Systeem is ongewijzigd.	
<b>Alternatief scenario trigger (A2):</b> In stap 3 van het main scenario. [Actor selecteert een boeking die niet bestaat]  <b>Alternatief scenario stappen(A2):</b> <ol style="list-style-type: none"><li>1. Systeem toont foutmelding.<ol style="list-style-type: none"><li>1.1 Systeem begint opnieuw met stap 1 van het main scenario.</li></ol></li></ol>	



## Use case 5: QR-Code scannen

<b>ID:</b> 5	<b>Naam:</b> QR-Code scannen
<b>Actor:</b> Gebruiker	
<b>Samenvatting:</b> Systeem vraagt om <b>QR-code(in)</b> . Actor laat de QR-code scannen. Systeem bevestigt de QR-code en opent de deur.	
<b>Precondities:</b> Geen	
<b>Main scenario:</b> <ol style="list-style-type: none"><li>1. Systeem vraagt om QR-code.</li><li>2. Actor laat de QR-code scannen.</li><li>3. Systeem bevestigt de QR-code en laat groen beeldscherm zien.</li></ol>	
<b>Postcondities:</b> Geen	
<b>Alternatief scenario trigger (A1):</b> In stap 2 van het main scenario. [Actor annuleert use case.] <b>Alternatief scenario stappen(A1):</b> <ol style="list-style-type: none"><li>1. Systeem beëindigt de use case.</li></ol> <b>Post conditie:</b> Systeem is ongewijzigd.	
<b>Alternatief scenario trigger (A2):</b> In stap 3 van het main scenario. [QR-code is niet geldig] <b>Alternatief scenario stappen(A2):</b> <ol style="list-style-type: none"><li>1. Systeem toont rood scherm.<ol style="list-style-type: none"><li>1.1 Systeem begint opnieuw met stap 1 van het main scenario.</li></ol></li></ol>	

## Use case 6: Account verifiëren

In en outs van verificatiecode vragen. (is het een out als deze aan de student wordt gevraagd?  
En een in als die gegenereerd wordt door het systeem?)

In en outs van studentenmail vragen (als deze attribuut wordt gebruikt om een email te versturen is het een out?)

<b>ID:</b> 6	<b>Naam:</b> Account verifiëren
<b>Actor:</b> Gebruiker	
<b>Samenvatting:</b> Het systeem vraagt om een <b>studentnummer(in), voornaam(in), achternaam(in), studentenmail(in)</b> . De actor verstrekt deze gegevens. Het systeem vraagt om een bevestiging en de actor bevestigt. Het systeem genereert een verificatiecode en slaat deze op, het systeem stuurt een email met verificatiecode naar de studentenmail en vraagt om de verificatiecode. Actor vult deze gegevens in. Systeem verifieert het account in het systeem.	
<b>Precondities:</b> Ingelogd met gmail account.	
<b>Main scenario:</b> <ol style="list-style-type: none"><li>1. Het systeem vraagt om een <b>studentnummer(in), voornaam(in), achternaam(in), studentenmail(in)</b>.</li><li>2. De actor verstrekt deze gegevens.</li><li>3. 3.1 Het systeem genereert een verificatiecode en slaat deze op. 3.2 Het systeem stuurt een email met verificatiecode naar de studentenmail. 3.3 Het systeem vraagt om de verificatiecode.</li><li>4. Actor vult deze gegevens in.</li><li>5. Systeem verifieert het account in het systeem.</li></ol>	
<b>Postcondities:</b> Account is geverifieerd.	
<b>Alternatief scenario trigger (A1):</b> In stap 2, 4 van het main scenario. [Actor annuleert usecase.]  <b>Alternatief scenario stappen(A1):</b> <ol style="list-style-type: none"><li>1. Systeem beëindigt de usecase.</li></ol> <b>Post conditie:</b> Systeem is ongewijzigd.	

**Alternatief scenario trigger (A2):**

In stap 2 van het main scenario.

[Actor voert niet alle gegevens in.]

**Alternatief scenario stappen(A2):**

1. Systeem toont foutmelding.

1.1 Systeem begint opnieuw met stap 1 van het main scenario.

**Alternatief scenario trigger(A3):**

Na stap 2 van het main scenario

[schoolmail in al in gebruik voor een account]

**Alternatief scenario stappen(A3):**

1.

1.1 Systeem toont melding dat schoolmail al in gebruik is.

1.2 Systeem systeem gaat verder met stap 1 van het main scenario

**Post conditie:**

Systeem is ongewijzigd.

# Wireframes

## Wireframe UC-01 (Ruimte boeken):

Student:

SDLab

Home Boekingsoverzicht Logout

October 2019

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

**Boeking plaatsen**

Datum

Begintijd

Eindtijd

Opmerking

**Boeken**

Een student kan een ruimte boeken door op bovenstaand scherm de 4 inputvelden in te voeren. Als de student dit gedaan heeft en op boeken heeft geklikt is er een aanvraag gestuurd voor deze boeking.

## Beheerder:

SDLab

https://www.sdlab.nl

**SDLab**   [Home](#)   [Boekingsoverzicht](#)   [Boeking beoordelen](#)   [Logout](#)

<

October 2019

>

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

**Boeking plaatsen**

Datum

Begintijd

Eindtijd

Opmerking

Boeken

Een beheerder kan een ruimte boeken door op bovenstaand scherm de 4 inputvelden in te voeren. Als de beheerder dit gedaan heeft en op boeken heeft geklikt is er een aanvraag gestuurd voor deze boeking.

## Wireframe UC-02 (Boeking beoordelen):

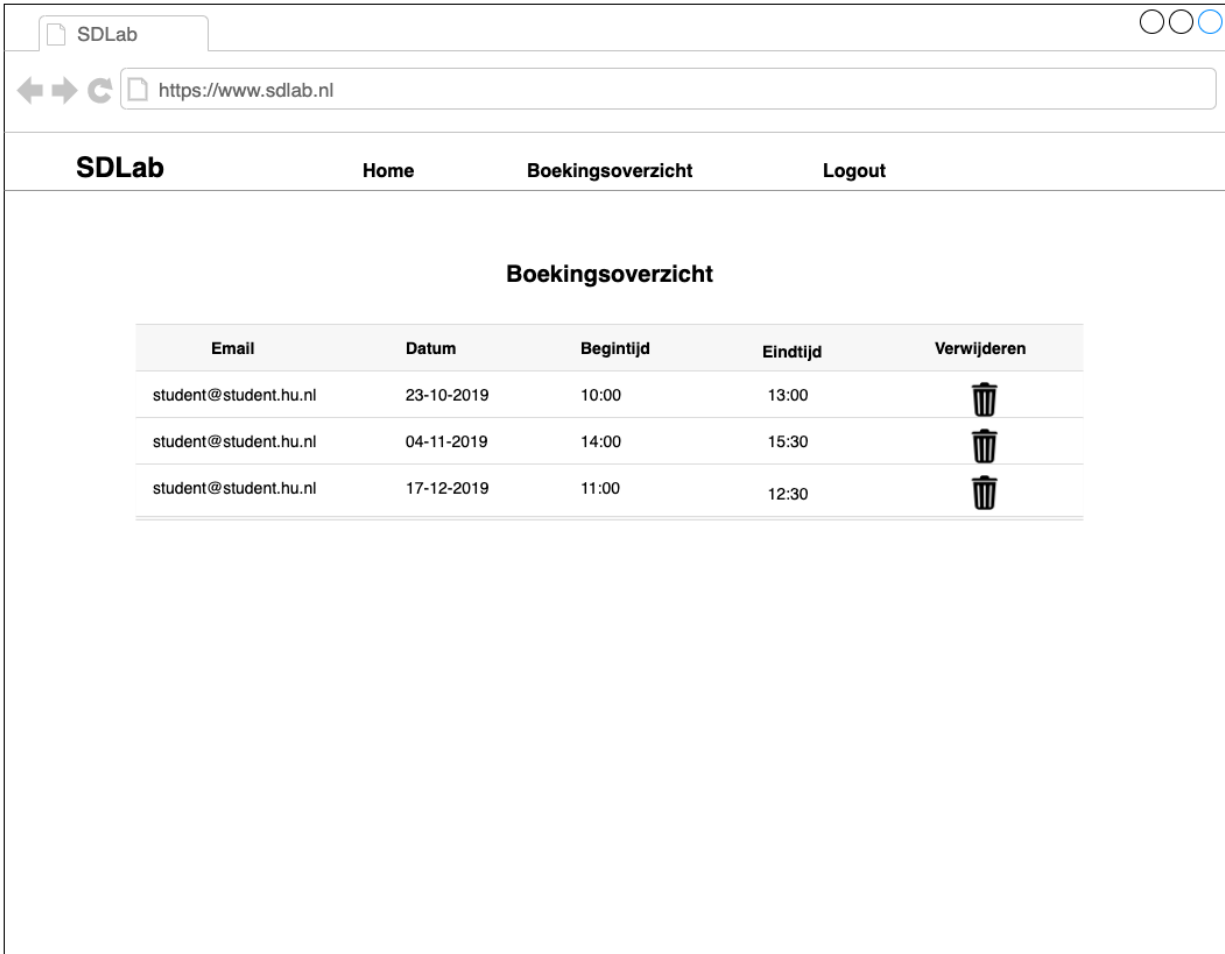
Beheerder:

Email	Studentennummer	Voornaam	Achternaam	Datum	Begintijd	Eindtijd	Goedkeuren	Afkeuren
student@student.hu.nl	123456	Bob	Ross	23-10-2019	10:00	13:00	✓	
student@student.hu.nl	654321	Bob	Ross	04-11-2019	11:30	12:30	✓	
student@student.hu.nl	321456	Bob	Ross	17-12-2019	14:00	15:45	✓	

Een beheerder kan via de navbar bij de pagina boeking beoordelen komen. Hier ziet de beheerder boekingen die in afwachting zijn, deze kan de beheerder goedkeuren door op het vinkje te klikken of afkeuren door op het vuilnisbakje te klikken.

## Wireframe UC-03 (Boeking bekijken):

Student:



De student kan op de boekingsoverzicht pagina een overzicht zien van de boekingen die deze student gedaan heeft.

## Beheerder:

SDLab

https://www.sdlab.nl

SDLab




Home

Boekingsoverzicht

Boeking beoordelen

Logout

### Boekingsoverzicht

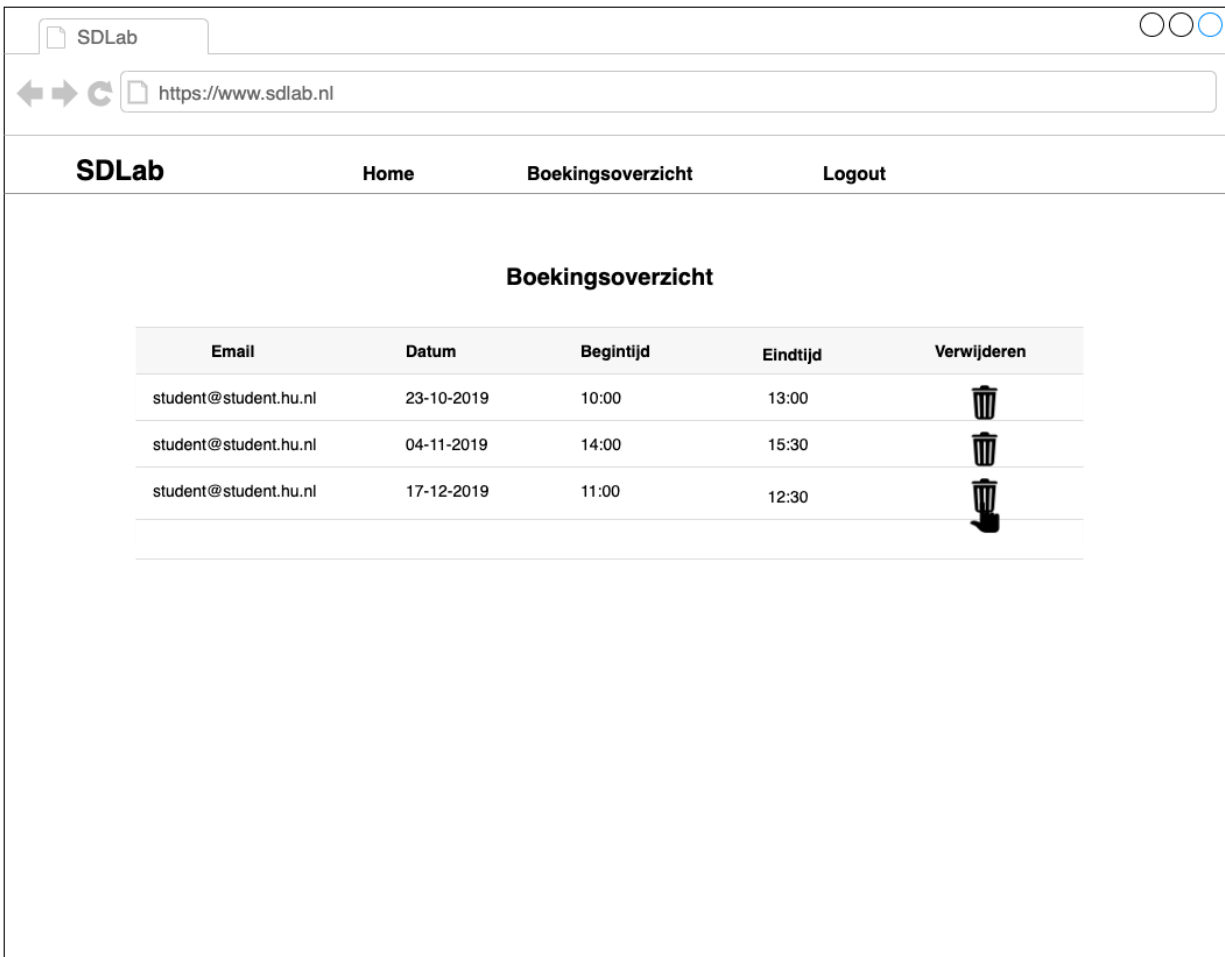
Email	Datum	Begintijd	Eindtijd	Verwijderen
student@student.hu.nl	23-10-2019	10:00	13:00	
student@student.hu.nl	04-11-2019	14:00	15:30	
student@student.hu.nl	17-12-2019	11:00	12:30	

De beheerder kan op de boekingsoverzicht pagina de boekingen inzien die de beheerder gedaan heeft.



## Wireframe UC-04 (Boeking verwijderen):

Student:



De student kan via de boekingsoverzicht pagina zijn boekingen verwijderen door op het vuilnisbakje te klikken.

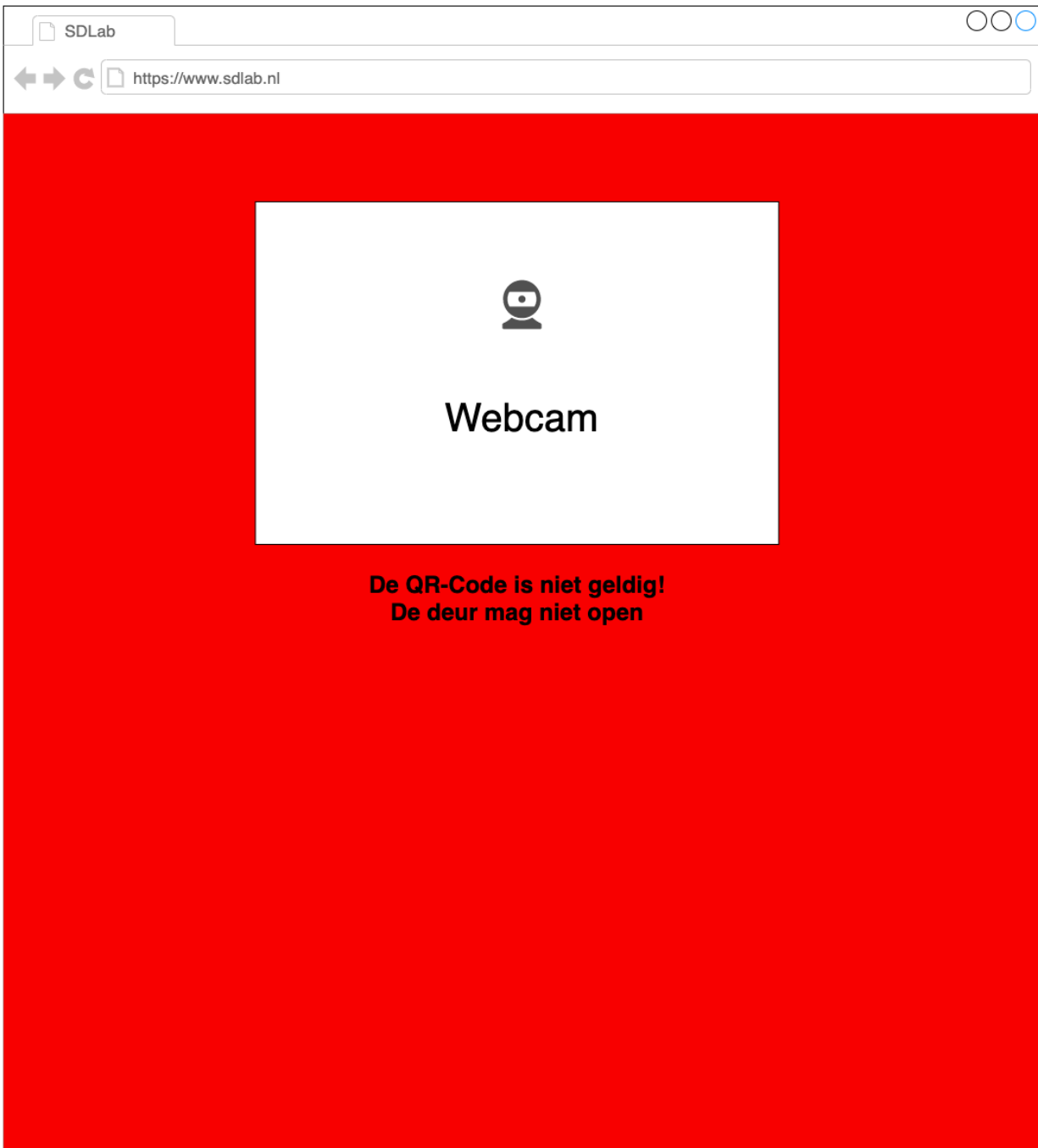
## Beheerder:

The screenshot shows a web browser window with the address bar displaying `https://www.sdlab.nl`. The page has a navigation bar with the following links: **SDLab**, **Home**, **Boekingsoverzicht** (active), **Boeking beoordelen**, and **Logout**. The main content area is titled **Boekingsoverzicht** and contains a table with booking details. The table has five columns: **Email**, **Datum**, **Begintijd**, **Eindtijd**, and **Verwijderen**. There are three rows of booking data, each with a trash can icon in the **Verwijderen** column. A mouse cursor is pointing at the trash can icon of the third row.

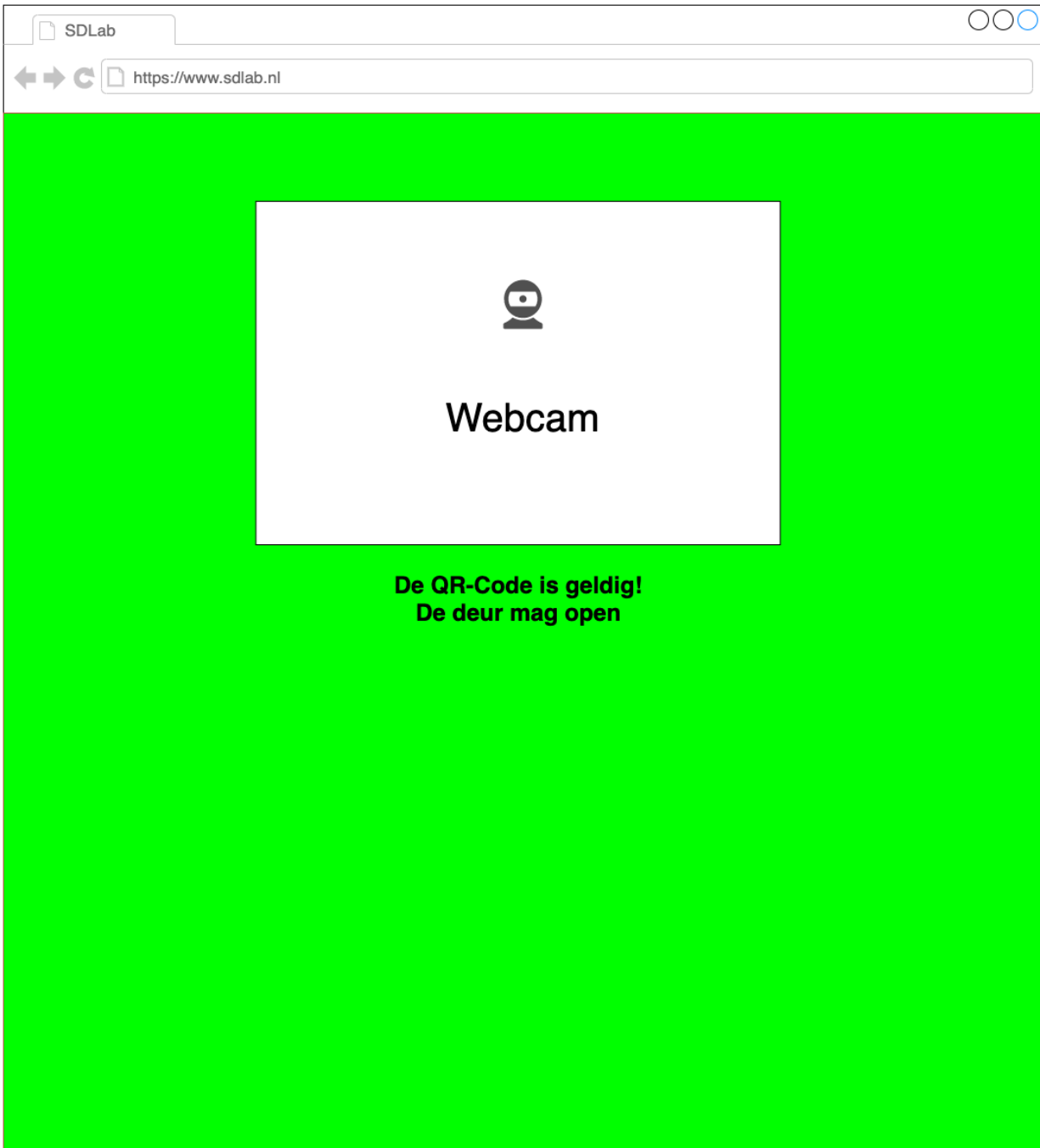
Email	Datum	Begintijd	Eindtijd	Verwijderen
student@student.hu.nl	23-10-2019	10:00	13:00	
student@student.hu.nl	04-11-2019	14:00	15:30	
student@student.hu.nl	17-12-2019	11:00	12:30	

De beheerder kan op de boekingsoverzicht pagina de boekingen verwijderen die de beheerder gemaakt heeft door op het vuilnisbakje te klikken.

## Wireframe UC-05 (QR-Code scannen):



Op de plek van de QR-Code komt het beeld van de webcam waarop je de QR-Code kan scannen. Als in dit geval de QR-Code niet geldig is wordt het scherm rood en mag de deur niet worden opgemaakt.



Als na het scannen van de QR-Code het scherm groen word mag de deur wel worden opgemaakt.

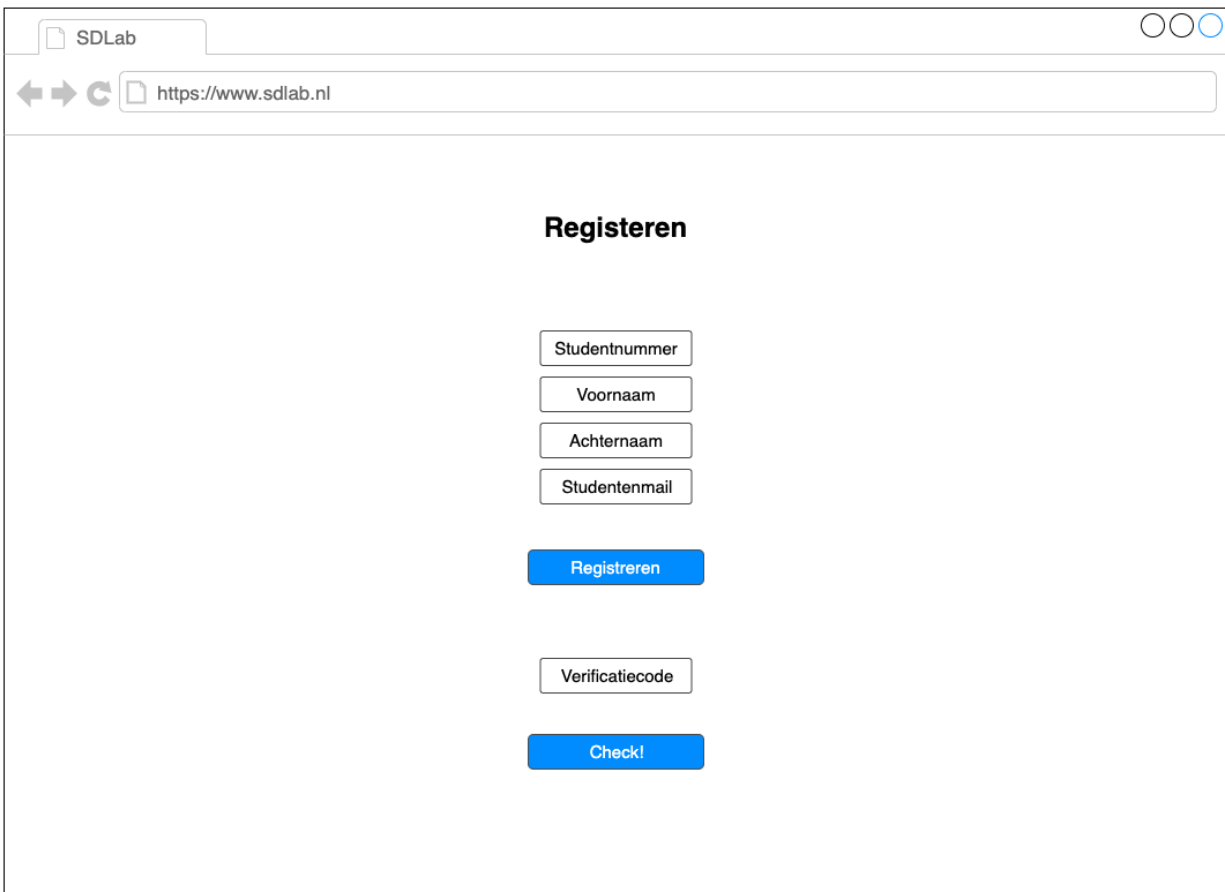
## Wireframe UC-06 (Account verifiëren):

Gebruiker registreert zich op de website:

The wireframe shows a web browser window with a single tab labeled 'SDLab'. The address bar contains the URL 'https://www.sdlab.nl'. The main content area features a centered heading 'Registeren'. Below the heading are four stacked text input fields labeled 'Studentnummer', 'Voornaam', 'Achternaam', and 'Studentenmail'. At the bottom of the form is a blue button labeled 'Registreren'.

De gebruiker komt na het inloggen met zijn/haar Google Account terecht op de registratiepagina als de gebruiker nog niet bekend is in het systeem. Hier vult de gebruiker zijn/haar studentnummer, voornaam, achternaam en studentenmail in. Daarna klikt de gebruiker op de knop registreren, hierna ontvangt de gebruiker op zijn/haar studentenmail een verificatiecode.

## Verificatiecode invoeren:



The screenshot shows a web browser window with the title 'SDLab' and the address bar displaying 'https://www.sdlab.nl'. The main content area is titled 'Registeren' and contains a registration form. The form consists of four input fields stacked vertically: 'Studentnummer', 'Voornaam', 'Achternaam', and 'Studentenmail'. Below these fields is a blue button labeled 'Registreren'. Further down, there is another input field labeled 'Verificatiecode' and a blue button labeled 'Check!'.

**Registeren**

Studentnummer

Voornaam

Achternaam

Studentenmail

Registreren

Verificatiecode

Check!

Nadat de gebruiker op de knop registreren geklikt heeft komt er een nieuw inputveld voor de verificatiecode, hier kan de gebruiker de toegestuurde code invoeren en zijn/haar account verifiëren.