

# EasyBook - Requirements Definition

*A booking system for the Västerås Flygmuseum*

<b>Version</b>	<b>Type</b>	<b>Date</b>	<b>Name</b>	<b>Description</b>
0.1	Draft		Robert Engelmann	First draft
0.2	Update	2014-11-12	Sebastian Kunze	Added Use Cases
0.3	Update	2014-11-12	Sebastian Kunze	Added use Case Diagrams
1.0	Review	2014-11-13	Sebastian Kunze	Finalized latest version
1.1	Update	2014-11-23	Robert Engelmann	Added feedback from customer
1.2	Update & Review	2014-11-26	Sebastian Kunze	Updated non-functional requirements and reviewed feedback from customer
1.3	Update	2014-11-29	Robert Engelman	Renamed the role "user" to "customer" in textual contexts.

## **Table of Contents**

- [1. Description and Definitions](#)
  - [1.1. General Description](#)
  - [1.2 Actors](#)
- [2 Functional requirements](#)
  - [2.1 Use Cases](#)
    - [2.1.1 Customer](#)
      - [2.1.1.1 Book simulator](#)
      - [2.1.1.2 Cancel booking](#)
    - [2.1.2 Instructor](#)
      - [2.1.2.1 Log in](#)
      - [2.1.2.2 View booking](#)
      - [2.1.2.3 Assign booking to herself/himself](#)
      - [2.1.2.4 Prolong booking](#)
    - [2.1.3 Coordinator](#)
      - [2.1.3.1 Confirm booking](#)
      - [2.1.3.2 Create booking](#)
      - [2.1.3.3 Delete booking](#)
      - [2.1.3.4 Edit booking](#)
      - [2.1.3.5 Assign booking](#)
    - [2.1.4 Administrator](#)
      - [2.1.4.1 Create account](#)
      - [2.1.4.3 Disable account](#)
      - [2.1.4.3 Edit account](#)
      - [2.1.4.4 Change price](#)
      - [2.1.4.5 Change time](#)
  - [2.2 Use Case Diagrams](#)
    - [2.2.1 Customer](#)
    - [2.2.2 Instructor](#)
    - [2.2.3 Coordinator](#)
    - [2.2.4 Administrator](#)
- [3 Non-functional requirements](#)
- [4 Validation of Requirements](#)
- [5 User Stories / Appendix](#)

## **1. Description and Definitions**

### **1.1. General Description**

We are to develop a new booking system for the Västerås Flygmuseum. The aim is to provide an easy to use, intuitive interface to book the flight simulators, which are operated by the

museum. The system will unburden the staff and reduce bureaucratic effort, while at the same time provide more comfort for the museums customers and flight instructors.

The customers will be able to select their favorite flying times via a web interface. If it is outside of the normal opening hours, the coordinator will arrange an instructor and confirm the booking with the customer. The instructors will have access to all the booking data, so they are always up to date and know what's going on. The people operating the cash point can also create bookings for people who are spontaneously visiting the museum. The system will be deployed on a server in the museum and available over the internet. In this way we will ensure an undisturbed flow of information.

The following sections contain detailed information on the actors, functional requirements (in form of Use Cases) and non-functional requirements. Additionally, a list of all User Stories from the Product Backlog is attached.

## 1.2 Actors

**Customer** A customer of the systems is a customer of the museum who wants to fly one of the museum's simulators. To fly a simulator, the customer needs to reserve it beforehand. During the opening hours of the museum, the customer simply books an available slot for a certain simulator. Outside the opening hours of the museum, the customer needs to request a simulator for a specific timespan. This should be as easy as possible and not be constrained by any registration process.

Following Use Cases can be performed by the customer:

- Book simulator
- Cancel booking

**Instructor** An instructor is a member of the museum and gives instructions to assigned customers that fly a particular simulator. The system provides the instructor an overview of today's booking for all simulators. If a customer wants to fly a simulator for some more time, the instructor can edit his assigned booking for today.

Following Use Cases can be performed by the instructor:

- Log in
- View booking
- Assign booking to herself/himself
- Prolong booking

**Coordinator** A coordinator takes booking requests by mail or phone and at the desk and adds them to the system. They have an overview of all bookings including the personal data and are also able to edit and even to delete bookings, if required. Whenever a customer requests a slot outside the opening hour of the museum, it is the coordinator's job to confirm or deny it. He can assign

an instructor who is being notified then.

Following Use Cases can be performed by the coordinator:

- Log in
- View booking
- Assign booking
- Prolong booking
- Confirm booking
- Create booking
- Delete booking
- Edit booking

### **Admin**

An administrator takes care of all administrative tasks. He manages the system's accounts and is able to change the system's variables.

Following Use Cases can be performed by the administrator:

- Log in
- View booking
- Assign booking
- Prolong booking
- Confirm booking
- Create booking
- Delete booking
- Edit booking
- Create account
- Disable account
- Edit account
- Change price
- Change time

## 2 Functional requirements

The following functional requirements describe the system's behaviour in respect to the Västerås Flygmuseum:

### 2.1 Use Cases

#### 2.1.1 Customer

A customer is someone who wants to fly one of the museum's simulators.

##### 2.1.1.1 Book simulator

<b>Use Case</b>	Book simulator
<b>Actor</b>	System, Customer, Coordinator
<b>Precondition</b>	/
<b>Main path:</b>	<ol style="list-style-type: none"><li>1. Customer selects new booking on website.</li><li>2. Customer selects a simulator.</li><li>3. Customer sees the simulator's availability.</li><li>4. Customer selects a free time slot during the opening hours of the museum.</li><li>5. Customer enters her/his name, email address and phone number.</li><li>6. Customer provides additional information.</li><li>7. Customer sees pricing information.</li><li>8. Customer sends all booking information.</li><li>9. System verifies provided information.</li><li>10. System sends notification to customer.</li></ol>
<b>Alternative path (A1):</b>	<ol style="list-style-type: none"><li>3. Customer selects time span outside the opening hours of the museum</li></ol>
<b>Alternative path (A2):</b>	<ol style="list-style-type: none"><li>9. System verifies provided information.<ol style="list-style-type: none"><li>a. System reminds customer that its name, email address, phone number or postal address is incorrect.</li><li>b. Customer corrects name, email address, phone number or postal address accordingly.</li><li>c. Customer sends all booking information again.</li></ol></li></ol>
<b>Alternative path (A3):</b>	<ol style="list-style-type: none"><li>10. System send confirmation to coordinator.<ol style="list-style-type: none"><li>a. Coordinator confirms booking.</li><li>b. System sends notification to customer.</li></ol></li></ol>

<b>Alternative Path (A4)</b>	5. Customer enters her/his name, email address and phone number as well as a postal address for the corresponding invoice.
------------------------------	--

#### 2.1.1.2 Cancel booking

<b>Use Case</b>	Cancel booking
<b>Actor</b>	System, Customer
<b>Precondition</b>	Book simulator
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Customer selects <i>cancel booking</i> on website.</li> <li>2. Customer enters booking ID and her/his name</li> <li>3. Customer clicks <i>cancel</i></li> <li>4. System sends notification to use</li> </ol>
<b>Alternative path (A1):</b>	<ol style="list-style-type: none"> <li>2. Customer forgot her/his booking ID <ol style="list-style-type: none"> <li>a. Customer calls museum</li> <li>b. Coordinator cancels booking</li> </ol> </li> </ol>

#### 2.1.2 Instructor

An instructor is a trained staff member of the museum who assists a customer during his simulator flight.

##### 2.1.2.1 Log in

<b>Use Case</b>	Log in
<b>Actor</b>	System, Instructor
<b>Precondition</b>	/
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Instructor selects <i>log in</i> on the website.</li> <li>2. Instructors enters her/his username and password.</li> <li>3. Instructor clicks <i>log in</i>.</li> <li>4. System authenticates instructor.</li> <li>5. Systems forwards instructor to the <i>calendar</i>.</li> </ol>
<b>Alternative path (A1):</b>	<ol style="list-style-type: none"> <li>2. Instructor forgot her/his password <ol style="list-style-type: none"> <li>a. Instructor clicks <i>forgot password</i></li> <li>b. Administrator resets password</li> </ol> </li> </ol>

##### 2.1.2.2 View booking

<b>Use Case</b>	View bookings
<b>Actor</b>	Instructor

<b>Precondition</b>	<ul style="list-style-type: none"> <li>● Log in</li> <li>● Book simulator / Create booking</li> </ul>
<b>Main path:</b>	1. Instructor clicks <i>bookings</i> on website.

#### 2.1.2.3 Assign booking to herself/himself

<b>Use Case</b>	Assign booking to herself/himself
<b>Actor</b>	System, Instructor, Customer
<b>Precondition</b>	<ul style="list-style-type: none"> <li>● Log in</li> <li>● Book simulator / Create booking</li> <li>● View bookings</li> </ul>
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Instructor clicks on a highlighted unassigned booking.</li> <li>2. Instructor sees date, time span, name, email address, phone number and postal address of the customer.</li> <li>3. Instructor clicks <i>assign to me</i>.</li> </ol>
<b>Alternative path (A1):</b>	<ol style="list-style-type: none"> <li>3. Instructor clicks <i>assign to me</i>. <ol style="list-style-type: none"> <li>a. Systems reminds instructor that she/he is already assigned to another booking.</li> <li>b. Instructor confirms her/his change.</li> <li>c. System removes instructor from the current booking.</li> <li>d. System assigns instructor to the chosen booking.</li> </ol> </li> </ol>

#### 2.1.2.4 Prolong booking

<b>Use Case</b>	Prolong booking
<b>Actor</b>	System, Instructor
<b>Precondition</b>	<ul style="list-style-type: none"> <li>● Book simulator</li> <li>● Assign booking</li> <li>● View bookings</li> </ul>
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Instructor selects her/his currently ongoing booking.</li> <li>2. Instructor clicks <i>prolong</i>.</li> <li>3. Instructor selects end time.</li> </ol>
	2. Instructor clicks <i>prolong</i> .

	<ol style="list-style-type: none"> <li>a. System reminds instructor that she/he is already assigned to an another booking.</li> <li>b. Instructor confirms prolonging the current booking.</li> </ol>
--	---

### 2.1.3 Coordinator

A coordinator is a staff member of the museum who manages simulator bookings.

#### 2.1.3.1 Confirm booking

<b>Use Case</b>	Confirm booking
<b>Actor</b>	System, Coordinator, Instructor, Customer
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Log in</li> <li>• Book simulator / Create booking</li> <li>• View bookings</li> </ul>
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Coordinator clicks on a highlighted unconfirmed booking outside the opening hours of the museum.</li> <li>2. Coordinator sees date, time span, name, email address, phone number and postal address of customer.</li> <li>3. Coordinator clicks <i>confirm</i>.</li> <li>4. System notifies customer.</li> </ol>
<b>Alternative path (A1):</b>	<ol style="list-style-type: none"> <li>3. Coordinator assigns an instructor.</li> <li>4. Coordinator clicks <i>confirm</i>.</li> <li>5. System notifies instructor and customer.</li> </ol>

#### 2.1.3.2 Create booking

<b>Use Case</b>	Create booking
<b>Actor</b>	Customer, Instructor, Coordinator
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Log in</li> </ul>
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Coordinator clicks <i>new booking</i> on website.</li> <li>2. Coordinator enters name, email address, phone number and postal address of the customer.</li> <li>3. Coordinator provides additional information.</li> <li>4. Coordinator sees pricing information</li> <li>5. Coordinator sends booking</li> <li>6. System verifies information</li> <li>7. System sends notification</li> </ol>



<b>Alternative path (A1):</b>	<ol style="list-style-type: none"> <li>5. Coordinator assigns instructor</li> <li>6. Coordinator sends booking</li> <li>7. System verifies information</li> <li>8. System sends notification</li> </ol>
-------------------------------	---

#### 2.1.3.3 Delete booking

<b>Use Case</b>	Delete booking
<b>Actor</b>	System, Instructor, Coordinator
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Log in</li> <li>• Book simulator / Create booking</li> <li>• View booking</li> </ul>
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Coordinator selects a booking.</li> <li>2. Coordinator clicks <i>delete</i>.</li> <li>3. System shows a pop-up that says <i>are you sure that you want to delete this booking</i>.</li> <li>4. Coordinator clicks <i>yes</i>.</li> <li>5. System notifies the customer</li> </ol>
<b>Alternative path (A1):</b>	<ol style="list-style-type: none"> <li>5. System notifies the customer and the assigned instructor</li> </ol>

#### 2.1.3.4 Edit booking

<b>Use Case</b>	Edit booking
<b>Actor</b>	System, Coordinator
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Log in</li> <li>• Book simulator / Create booking</li> <li>• View bookings</li> </ul>
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Coordinator selects a booking.</li> <li>2. Coordinator changes name, email address, phone number and/or postal address.</li> <li>3. Coordinator clicks <i>confirm</i>.</li> <li>4. System validates provided information.</li> </ol>
<b>Alternative path (A1):</b>	<ol style="list-style-type: none"> <li>2. Coordinator changes date and/or time <ol style="list-style-type: none"> <li>a. System notifies the customer and the assigned instructor</li> </ol> </li> </ol>
<b>Alternative path (A2):</b>	<ol style="list-style-type: none"> <li>4. System verifies provided information. <ol style="list-style-type: none"> <li>a. System remind coordinator that its name,</li> </ol> </li> </ol>

	<p>email address, phone number and/or postal address is incorrect.</p> <p>b. Coordinator corrects name, email address, phone number and/or postal address accordingly.</p> <p>c. Coordinator clicks <i>confirm</i> again.</p>
--	---

#### 2.1.3.5 Assign booking

<b>Use Case</b>	Assign booking
<b>Actor</b>	System, Instructor, Coordinator
<b>Precondition</b>	<ul style="list-style-type: none"> <li>● Log in</li> <li>● Book simulator / Create booking</li> <li>● View bookings</li> </ul>
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Coordinator selects a booking.</li> <li>2. Coordinator chooses an instructor from the drop-down menu.</li> <li>3. Coordinator clicks <i>confirm</i>.</li> <li>4. System notifies the instructor.</li> </ol>
<b>Alternative path (A1):</b>	<ol style="list-style-type: none"> <li>3. Coordinator clicks <i>confirm</i>. <ol style="list-style-type: none"> <li>a. System reminds the coordinator that chosen instructor is already assigned to an another booking.</li> <li>b. Coordinator confirms her/his change.</li> <li>c. System removes instructor from the current booking.</li> <li>d. System assigns instructor to chosen booking.</li> </ol> </li> </ol>

#### 2.1.4 Administrator

A administrator is a special staff member of the museum who manages all functionality and variables of the system.

##### 2.1.4.1 Create account

<b>Use Case</b>	Create account
<b>Actor</b>	System, Administrator
<b>Precondition</b>	<ul style="list-style-type: none"> <li>● Log in</li> </ul>
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Administrator clicks <i>settings</i> on website.</li> <li>2. Administrator clicks <i>accounts</i>.</li> </ol>

	<ol style="list-style-type: none"> <li>3. Administrator clicks <i>create</i>.</li> <li>4. Administrator provides name and email address.</li> <li>5. Administrator chooses a role.</li> <li>6. Administrator clicks <i>confirm</i>.</li> <li>7. System verifies name and email address.</li> </ol>
<b>Alternative path (A1):</b>	<ol style="list-style-type: none"> <li>7. System verifies name and email address. <ol style="list-style-type: none"> <li>a. System remind administrator that name and/or email address is incorrect.</li> <li>b. Administrator corrects name and/or email address accordingly.</li> <li>c. Administrator clicks <i>confirm</i> again.</li> </ol> </li> </ol>

#### 2.1.4.3 Disable account

<b>Use Case</b>	Disable account
<b>Actor</b>	Administrator
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Log in</li> </ul>
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Administrator clicks <i>settings</i> on website.</li> <li>2. Administrator clicks <i>accounts</i>.</li> <li>3. Administrator selects an account.</li> <li>4. Administrator clicks <i>disable</i>.</li> </ol>

#### 2.1.4.3 Edit account

<b>Use Case</b>	Edit account
<b>Actor</b>	System, Administrator
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Log in</li> </ul>
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Administrator clicks <i>settings</i> on website.</li> <li>2. Administrator clicks <i>accounts</i>.</li> <li>3. Administrator selects an account.</li> <li>4. Administrator edits name and/or email address.</li> <li>5. Administrator clicks <i>confirm</i>.</li> <li>6. System verifies name and email address.</li> </ol>
<b>Alternative path (A1):</b>	<ol style="list-style-type: none"> <li>6. System verifies name and email address. <ol style="list-style-type: none"> <li>a. System remind administrator that name and/or email address is incorrect.</li> <li>b. Administrator corrects name and/or email address accordingly.</li> <li>c. Administrator clicks <i>confirm</i> again.</li> </ol> </li> </ol>

#### 2.1.4.4 Change price

<b>Use Case</b>	Change price
<b>Actor</b>	Administrator
<b>Precondition</b>	<ul style="list-style-type: none"> <li>● Log in</li> </ul>
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Administrator clicks <i>settings</i>.</li> <li>2. Administrator clicks <i>simulators</i>.</li> <li>3. Administrator selects a simulator.</li> <li>4. Administrator changes price.</li> <li>5. Administrator clicks <i>confirm</i>.</li> <li>6. System verifies price.</li> </ol>
<b>Alternative path (A1):</b>	<ol style="list-style-type: none"> <li>4. Administrator changes price               <ol style="list-style-type: none"> <li>a. System remind administrator that price is incorrect.</li> <li>b. Administrator corrects price accordingly.</li> </ol> </li> </ol>
<b>Alternative path (A2):</b>	<ol style="list-style-type: none"> <li>6. System verifies price.               <ol style="list-style-type: none"> <li>a. System remind administrator that price is incorrect.</li> <li>b. Administrator corrects price accordingly.</li> <li>c. Administrator clicks <i>confirm</i> again.</li> </ol> </li> </ol>

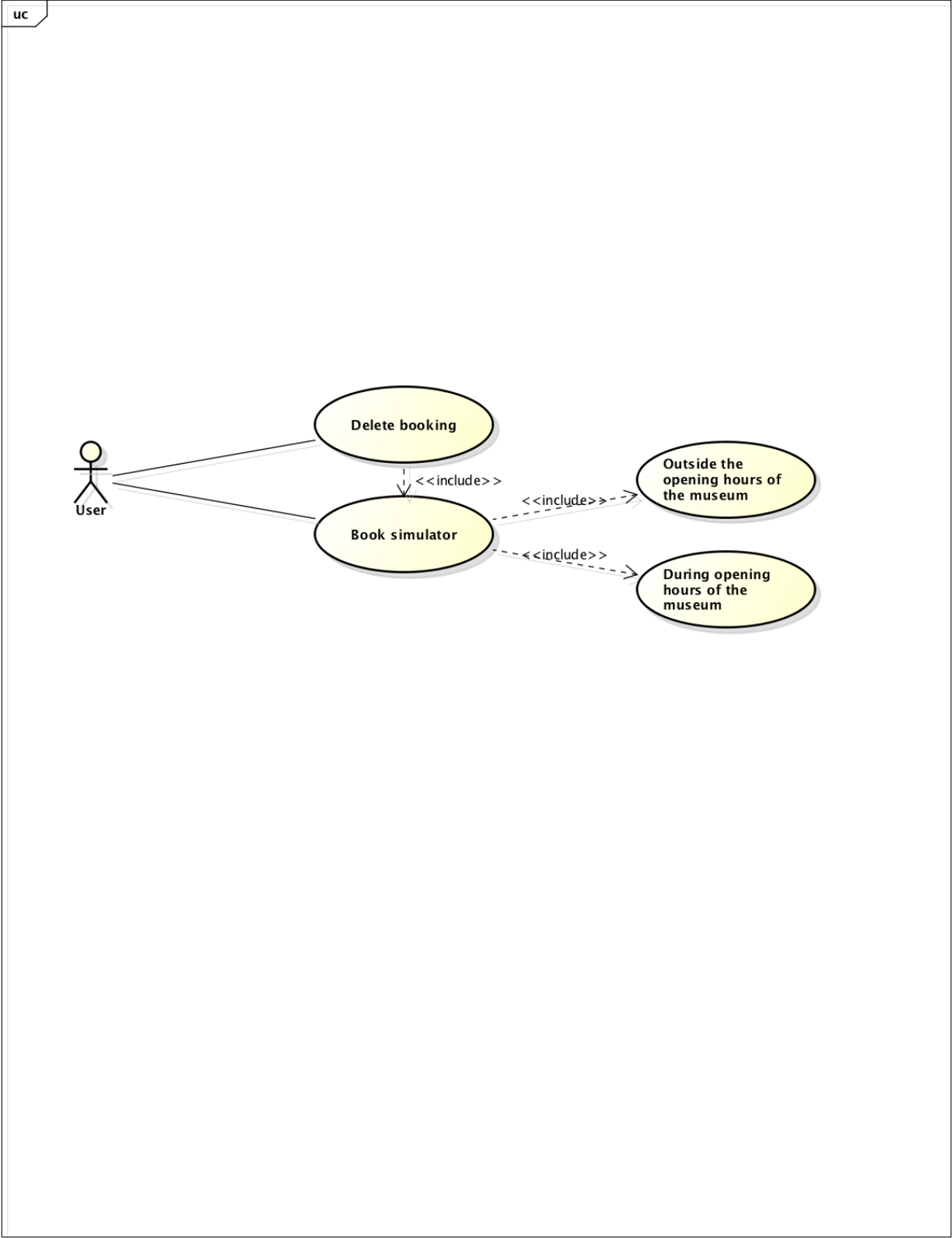
#### 2.1.4.5 Change time

<b>Use Case</b>	Change time
<b>Actor</b>	Administrator
<b>Precondition</b>	<ul style="list-style-type: none"> <li>● Log in</li> </ul>
<b>Main path:</b>	<ol style="list-style-type: none"> <li>1. Administrator clicks <i>settings</i>.</li> <li>2. Administrator clicks <i>timeslots</i></li> <li>3. Administrator clicks <i>add new day</i></li> <li>4. Administrator selects simulator and defines start and end points for time slots.</li> <li>5. Administrator clicks <i>confirm</i>.</li> </ol>
<b>Alternative path (A1):</b>	<ol style="list-style-type: none"> <li>4. Administrator clicks <i>copy from existing day</i>.               <ol style="list-style-type: none"> <li>a. Administrator selects existing day.</li> <li>b. Administrator clicks <i>confirm new day</i>.</li> </ol> </li> </ol>
<b>Alternative path (A2):</b>	<ol style="list-style-type: none"> <li>5. Administrator clicks <i>repeat weekly</i>.               <ol style="list-style-type: none"> <li>a. Administrator clicks <i>confirm</i>.</li> </ol> </li> </ol>

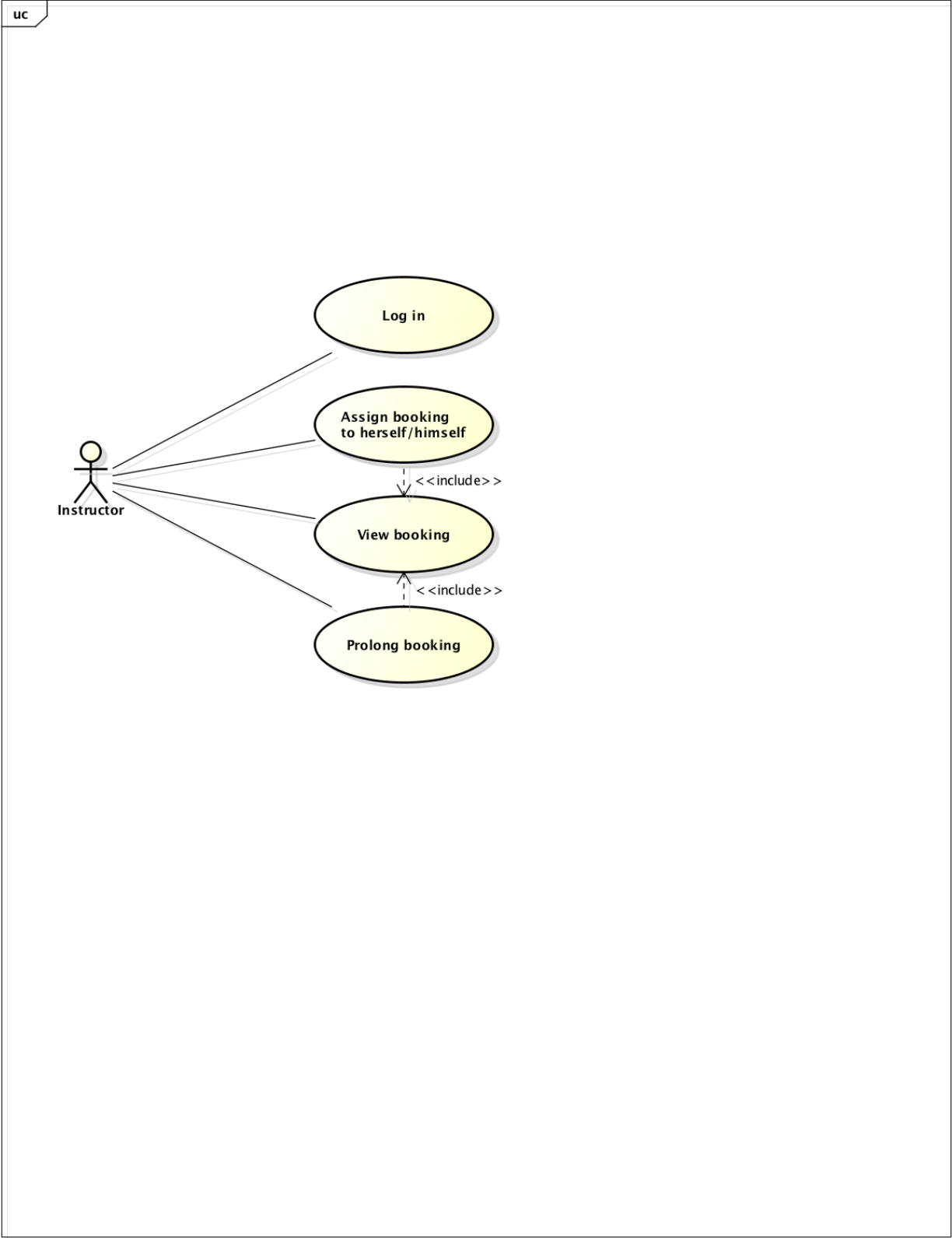
## 2.2 Use Case Diagrams

The following Use Case Diagrams are based on the Customer Cases in section [2.1](#). They represent the interaction of each actor from section [1.2](#) with the system in a simplified graphical representation.

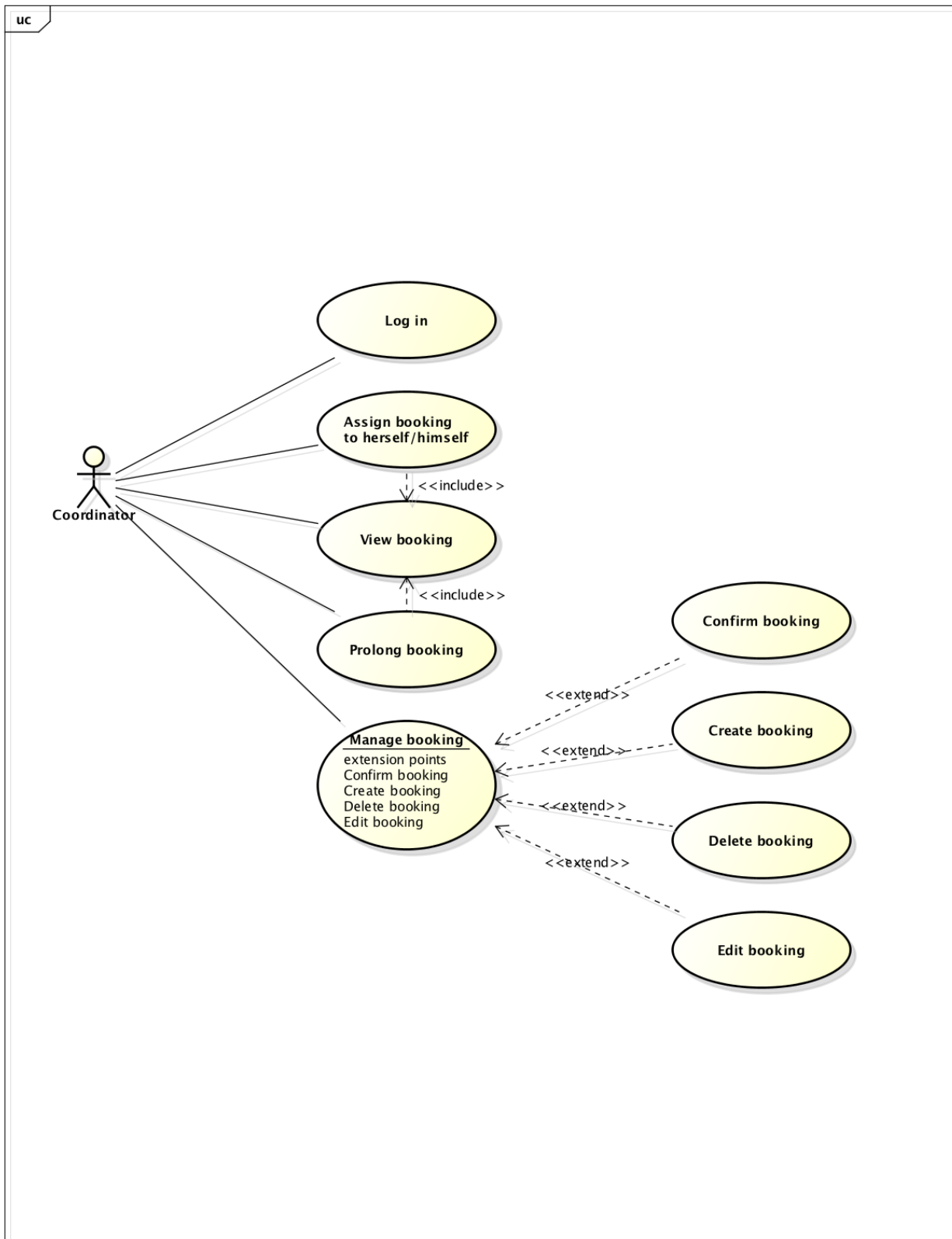
2.2.1 Customer



2.2.2 Instructor

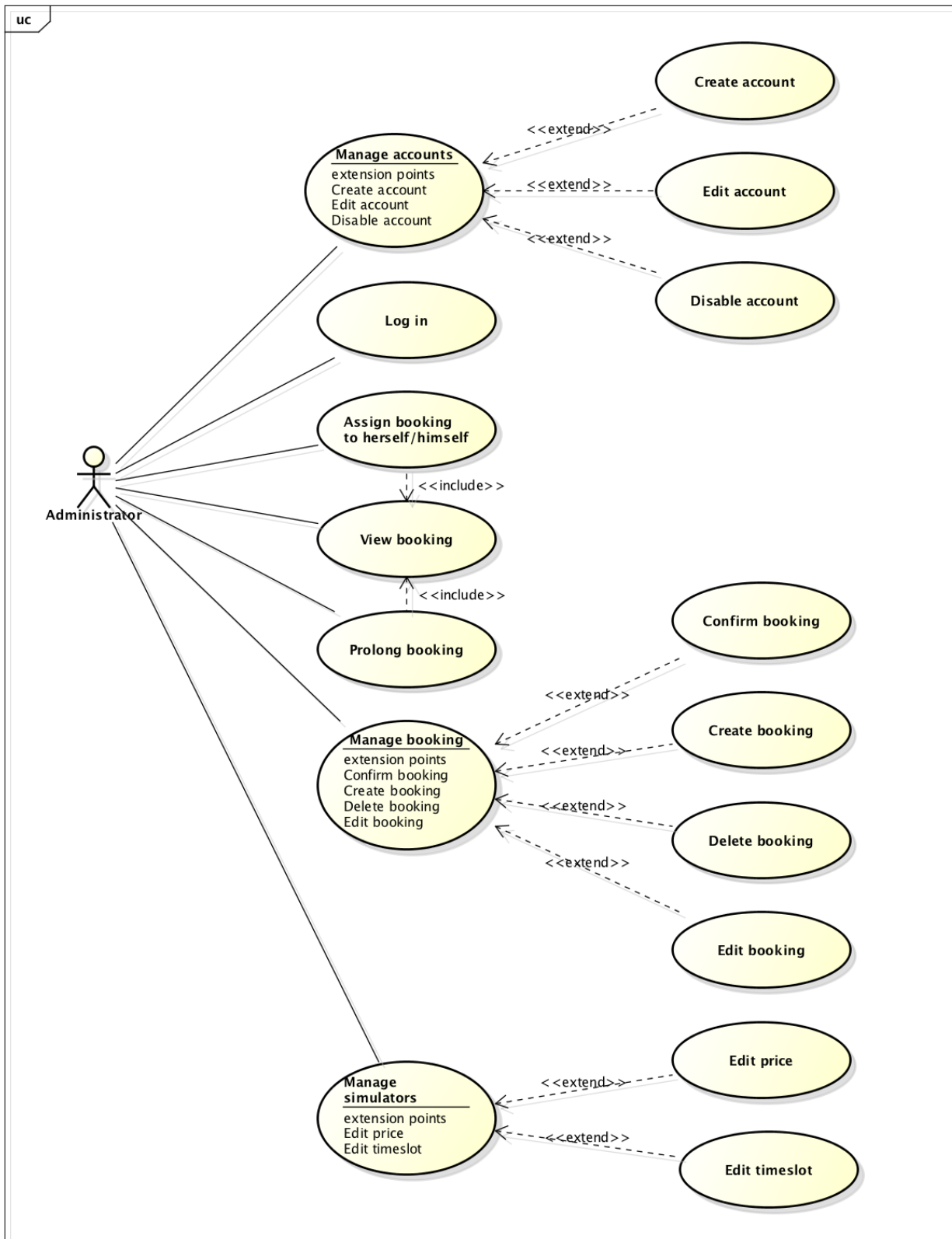


### 2.2.3 Coordinator





## 2.2.4 Administrator



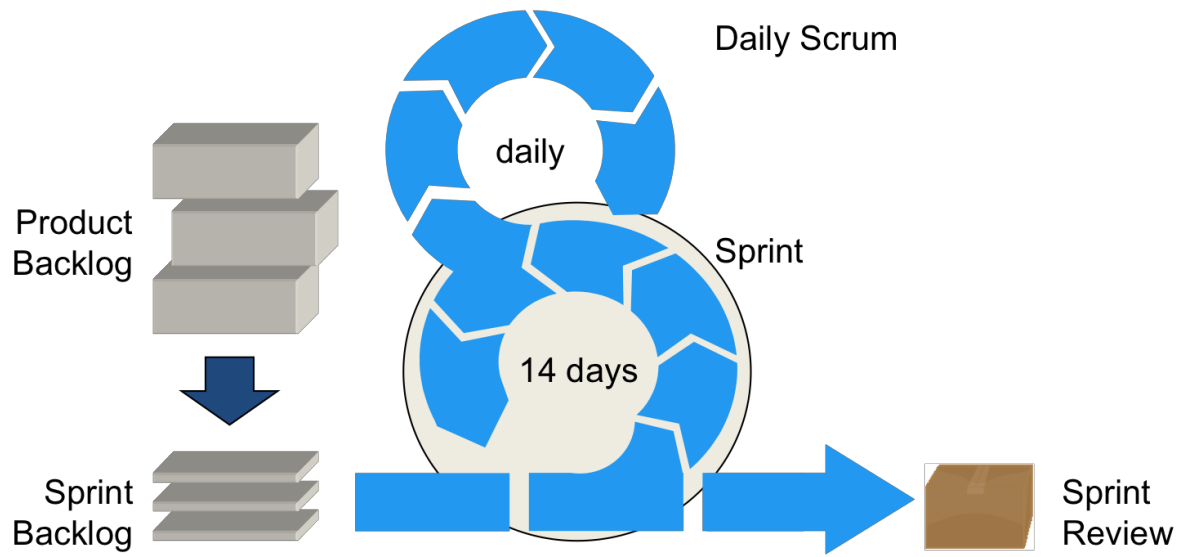
### 3 Non-functional requirements

Apart from the functional requirements, there are also some non-functional aspects that influence the system's architecture and design:

1. The system should provide an easy and simple overview for all users.
2. The system should observe all bookings and update itself.
  - The staff member always needs to be up to date and cannot miss any important new information.
3. The system should be easily usable and accessible. It will be orientated along the following guidelines:
  - a. Usability guideline: <http://guidelines.usability.gov/>
  - b. Accessibility guideline: <http://www.w3.org/WAI/eval/preliminary.html>
  - c. Validator: <http://validator.w3.org/>
  - d. Checklist: <http://www.useeffect.com/topic/25-point-website-usability-checklist>
    - The staff member's age ranges between 15 and 80+ years. This makes it extremely necessary to an intuitive graphical user interface.
4. The system should be maintainable and well documented. This includes usage of PHPDoc, well commented code, self explaining names and modularization of the components.
  - The Västerås Flygmuseum insists to use the system for the next years and want to be able to change it, whenever new requirements come up.
5. The system should not rely on uncertain and/or vague technologies.
6. The customer facing part of the system should be available in English and Swedish.
  - Many of the Västerås Flygmuseum's customers are tourists who don't speak swedish.
7. The system should be easily usable on mobile devices.
  - The Västerås Flygmuseum will eventually equip all instructors with tablets to ease the current workflow.
8. System must be licensed under the BSD license.

### 4 Validation of Requirements

In order to stay in contact with the Västerås Flygmuseum, Scrum is going to be used during the development period. Therefore, we are going to have a running systems once every two weeks. This running system is going to be used as a platform to review and validate the listed requirements in section [2](#). Those functional requirements might be adjusted according to the Västerås Flygmuseum and influence back the current version of the running system.



## 5 User Stories / Appendix

The following pages contain our in Asana maintained backlog, which contains our user stories.