

---

# User Requirements Specifications

## Business Networking and Mentorship Platform

---

**Author:** Desislav Hristov

**Class:** S2-CB02

**Last Edited:** 15.11.2023

**Version:** 2.0

---



## Contents

<b>1. Introduction:</b>	3
<b>2. Functional Requirements:</b>	3
2.1 User Profile Management:	3
2.2 Connection & Messaging:	3
2.3 Booking System:	3
2.4 Admin Oversight:	3
<b>3. Non-Functional Requirements:</b>	4
3.1 Performance:	4
3.2 Maintainability:	4
3.3 Security:	4
3.4 Scalability:	4
<b>Use Cases:</b>	4

## **1. Introduction:**

The Business Networking and Mentorship Platform is envisioned to be a hub for professionals. By bridging the gap between neophytes and experts, it promotes knowledge transfer, networking, and professional growth.

## **2. Functional Requirements:**

### **2.1 User Profile Management:**

- Profile Creation: Allow users to set up their accounts, specifying skills, experience, and goals.
- Profile Editing: Users can update their profiles, changing details or adding new credentials.

### **2.2 Connection & Messaging:**

- Search and Connect: Users can search for others based on specified criteria and send connection requests.
- Chat Interface: Once connected, users can send/receive messages in real-time.

### **2.3 Booking System:**

- Search Mentors: Mentees can search mentors based on industry, skills, or reviews.
- Schedule Sessions: Allows mentees to see a mentor's availability and book sessions.

### **2.4 Admin Oversight:**

- Monitoring: Admins can oversee platform activities, ensuring a positive environment.
- User Management: Admins have the power to modify or remove user accounts if required.

### 3. Non-Functional Requirements:

#### 3.1 Performance:

- Response Time: The system should respond quickly to user requests, avoiding long load times.
- Efficient Data Retrieval: Browsing, searching, or any database-reliant task should be seamless.

#### 3.2 Maintainability:

- Modular Design: The system's design should be modular, enabling easier updates or feature additions.

#### 3.3 Security:

- Authentication: Robust authentication mechanisms to prevent unauthorized access.

#### 3.4 Scalability:

- Load Balancing: As the platform grows, it should distribute traffic efficiently.

### Use Cases:

#### Use Case 1:

##### Book a Mentor Session

*Actor:* Mentee

*Main Success Scenario:*

1. Actor selects a mentor to book a session.
2. System displays mentor's time slots.
3. Actor chooses a desired time slot.
4. System confirms the booking, notifying both mentee and mentor.

#### **Extensions:**

3a: Time slot not available for the selected mentor.

.1: System suggests checking back later or choosing another mentor.

.2: End of use case.

## Use Case 2:

### Send a Message

*Actor:* User (Mentee/Mentor)

*Main Success Scenario:*

1. Actor selects another user to send a message.
2. System presents a messaging interface.
3. Actor types and sends the message.
4. System delivers the message to the recipient.

### **Extensions:**

2a: The selected user has disabled direct messaging.

.1: System informs the actor that the user can't receive direct messages.

.2: End of use case.

## Use Case 3:

### Admin Event Creation

*Actor:* Admin

*Main Success Scenario:*

1. Actor accesses the "Create Event" section.
2. System provides an event creation form.
3. Actor enters the event details and submits.
4. System validates, creates the event, and lists it for users.

### **Extensions:**

3a: Some essential event details are missing.

.1: System prompts admin to provide the missing information.

.2: Return to MSS step 3.

#### Use Case 4:

##### Submit a Review and Rating

Actor: Mentee

Main Success Scenario:

1. Actor completes a mentor session.
2. System prompts the actor to submit a review and rating.
3. Actor submits their feedback and selects a rating from 1 to 5.
4. System saves the review and updates the mentor's average rating.

##### **Extensions:**

3a: Actor chooses not to submit a review.

.1: System logs the completion of the session without a review.

.2: End of use case.

#### Use Case 5:

##### View Mentor Profiles

Actor: Mentee

Main Success Scenario:

1. Actor searches for mentors from the top rated mentors tab.
2. System displays a list of mentor profiles.
3. Actor selects a mentor to view detailed profile information.
4. System presents the selected mentor's full profile, including ratings and reviews.

##### **Extensions:**

2a: No mentors match the search criteria.

.1: System suggests adjusting the search parameters.

.2: End of use case.

### Use Case 6:

#### Manage Profile

Actor: User (Mentee/Mentor)

Main Success Scenario:

1. Actor selects to manage their profile.
2. System displays profile management options.
3. Actor updates their profile details.
4. System saves the changes and updates the profile.

#### **Extensions:**

3a: Invalid details entered.

.1: System prompts to correct the details.

.2: Return to MSS step 3.

### Use Case 7:

#### RSVP to an Event

Actor: User (Mentee/Mentor)

Main Success Scenario:

1. Actor browses upcoming events.
2. System displays event details.
3. Actor selects to RSVP to an event.
4. System confirms the RSVP and updates the event's attendee list.

#### **Extensions:**

2a: Event is full.

.1: System notifies that the event capacity has been reached.

.2: End of use case.

#### Use Case 8:

##### Cancel a Booking

Actor: Mentee

Main Success Scenario:

1. Actor selects a booked session.
2. System displays session details with a cancellation option.
3. Actor confirms cancellation.
4. System cancels the booking and notifies the mentor.

##### **Extensions:**

3a: Cancellation deadline passed.

.1: System informs about the inability to cancel after the deadline.

.2: End of use case.

#### Use Case 9:

##### Browse Mentorship Success Stories

Actor: User (Mentee/Mentor)

Main Success Scenario:

1. Actor selects the option to view mentorship success stories.
2. System displays a list of success stories shared by users.
3. Actor reads selected stories to gain insight and inspiration.

##### **Extensions:**

2a: No success stories are available.

.1: System suggests other areas of the platform to explore.

.2: End of use case.

#### Use Case 10:

##### Manage Messages

Actor: User (Mentee/Mentor)

Main Success Scenario:

Actor selects the messaging feature in the application.  
System displays the actor's inbox with a list of current conversations.  
Actor opens an existing conversation or starts a new one.



For a new conversation, the system prompts to select a recipient and compose a message.

Actor sends the message.

System delivers the message to the recipient and updates the conversation thread.

**Extensions:**

4a: Selected recipient has disabled direct messaging.

.1: System informs the actor that the recipient can't receive direct messages.

.2: End of use case.

4b: Actor decides not to start a new conversation.

.1: Actor returns to the inbox.

.2: End of use case.