

## Purpose

The results of the requirements elicitation and the analysis activities are documented in the Requirements Analysis Document (RAD). This document completely describes the system in terms of functional and nonfunctional requirements and serves as a contractual basis between the client and the developers.

## Audience

The audience for the RAD includes the client, the end users, the project manager, and the developers.

## Table of Contents

## Document History

Rev.	Author	Date	Changes
1	Kilian Northoff	1st July 2022	Wrote document

# 1 Introduction

The purpose of the Introduction is to provide a brief overview of the function of the system and the reasons for its development, its scope, and references to the development context. The introduction also includes the objectives and success criteria of the project.

## 1.1 Purpose of the System

Students want to participate in lectures and see announcements. They want to find fellow students with the same interests and share their opinion about lectures and course material. They also want to discuss exam questions and find the place where the exam takes place.

## 1.2 Scope of the System

This system should be able to be used everywhere both from mobile devices and PC.

# 2 Proposed System

## 2.1 Functional Requirements

### FR 1: Search for available lectures

A student can see all lectures of the current semester in his major and minor subject. He is able to join the lecture which saves it into his lecture list. He can also drop a lecture.

### FR 2: Check lecture details

A student can see details about a lecture such as the lecture times, the location of the lecture hall on a map and other lecture attendees including their name and picture.

### FR 3: Update profile

A student can update his profile settings and his profile picture. He can also change the notification settings.

### FR 4: Add comments

A student can add comments about a lecture and thus start a discussion. Others can like the comment and write follow-up comments.

### FR 5: Request friendship

A student can request friendship with another student who then receives a notification about the request. The second student can accept and reject friendship which both notify the first student.

### FR 6: Browse friend's lectures

A student can browse the lectures of his friends.

**FR 7: See announcements**

A student can see the lecture announcements and comment or like them.

**FR 8: Post updates to timeline**

A student can post updates to his timeline. Friends are notified about updates and can comment and like them. Certain updates are posted automatically such as saving a lecture into the lecture list or commenting in a lecture.

**FR 9: See lectures calendar**

A student can see all lectures in a calendar and save the events to the local calendar.

## 2.2 Non-Functional Requirements

Following NFR have to be taken into account.

### 2.2.1 Usability

The system should be intuitive to use, and the user interface should be easy to understand. All interactions should be completed in less than three clicks.

The design of the system should conform to the typical usability guidelines such as Nielsen's usability heuristics.

### 2.2.2 Implementation Requirements

A server subsystem with a couple of services must be used in the system.

## 2.3 System Models

The System models include scenarios, use cases, object model, and dynamic models for the system. This section should contain the complete functional specification, including mock-ups, paper-based prototypes or storyboards illustrating the user interface of the system and navigational paths representing the sequence of screens.

### 2.3.1 Scenarios

Arjun, a student from India, is studying computer science at TUM. He has business administration as minor subject. He is already used to visit the lectures in the FMI building in Garching from his first two semesters. The business administration lectures however, located in a lecture hall in the TUM city campus in Arcistrasse. He never visited the city campus before, so he does not know how to find the lecture halls for his minor subject. He browses through the lectures in the lecture catalog and finds the lecture "Foundations of Business Administration" with lecture times and the location of the lecture hall on a map. While he is attending the lecture, he makes contact with

fellow students who also attend the lecture and checks their comments. He likes one comment "Great exercises" by Jenny, who is also studying informatics. From Jenny's picture, he remembers that they met a week ago at the coffee machine in Garching. He requests friendship with Jenny (she might help him to pass the final) and adds a new comment about exam questions from earlier exams. While he is browsing, Jenny is notified about the friend request and accepts it. Arjun, in turn, is notified that Jenny has accepted his request and now browses through all the lectures that Jenny is visiting. This way he finds another interesting lecture "Cost Accounting" that he visits and saves it into his lecture list.

### 2.3.2 Use Case Model, Object Model and Dynamic Model

These can be found in the project in planning/diagram