

# **Software Requirements Specification JTracer**

by

**Albrit Bendo**

Version: 1.0

---

Date: May 17, 2021

---

**Contents**

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose . . . . .	1
1.2	Definitions and Acronyms . . . . .	1
<b>2</b>	<b>Requirements</b>	<b>2</b>
2.1	User Requirements . . . . .	2
2.2	System Requirements . . . . .	2
2.3	Functional Requirements . . . . .	2
2.4	Nonfunctional Requirements . . . . .	2
<b>3</b>	<b>System Design</b>	<b>3</b>
3.1	Use Case Diagram . . . . .	3
3.2	Sequence Diagram . . . . .	4
3.3	Class Diagram . . . . .	5
<b>4</b>	<b>Database Design</b>	<b>6</b>
4.1	ER Diagram . . . . .	6

# **1 Introduction**

JTracer is a web-based contact tracing application, developed especially for Jacobs University Bremen. Students and instructors are required to scan a QR-code which is attached to the door or in the entrance of a lecture hall. The contact details of the user are then forwarded to the university, which can manage them through the JTracer Access Management System (ACS). When there is an outbreak, this information will be managed by the respective admin. The records are stored in a user's private digital diary and also uploaded to the server. The app is also able to display a chosen classroom's current occupancy.

## **1.1 Purpose**

The purpose of this document is to specify the design of the JTracer development. It will explain the usage of the application, its requirements and constraints, its software design with supporting diagrams. This document is intended to serve as a guide for the implementation of the application.

## **1.2 Definitions and Acronyms**

### **Contact Tracing**

Contact tracing is a method in which public health authorities aim to restrict the spread of the virus by recognizing infected persons, disclosing their "contacts" to all individuals to whom they may have transmitted the disease, and, if possible, asking to quarantine infected individuals and their contacts. A contact may be someone who has been close to a highly infectious individual. Applications for proximity tracing will speed up contact tracing, using smartphones to quickly locate and inform contacts.

### **QR-Code**

QR is an abbreviation for "Quick Response." QR codes, despite their simplicity, are capable of storing a large amount of data. However, regardless of how much information they contain, when scanned, the QR code should allow the user to access information instantly – hence the name "Quick Response code."

### **Jacobs faculty and/or staff**

Jacobs University faculty and/or staff means a person which possesses and has an active email from the university.

### **External guest**

External guest is a person which is not part of the Jacobs University faculty and/or staff and does not possesses nor has an active email from the university.

## **2 Requirements**

### **2.1 User Requirements**

Assumed user is a student, professor, guest or participant of a hall.

- The user shall be able to authenticate using his/her Jacobs email if is a Jacobs faculty and/or staff member
- The user shall be able to authenticate using his/her contact information if is an external guest
- Logged in user shall be able to willingly check-in by scanning a QR-Code
- The user shall be able to confirm his/her presence in a hall after scanning a QR-Code
- The user shall be able to access a diary where a log of its history is kept (up to 14 days prior)
- The user shall be able to see how many people are attending the classroom
- The user shall be able to contact the admin

### **2.2 System Requirements**

- The system shall have a home page that provides all the functionality needed
- The system shall be accessible without an installation
- The system shall be secure with only hashed password saving and encryption
- The system shall be restricted to university emails authentication
- The system shall provide a manual check-in for external guests
- The system shall provide how many people are attending the classroom

### **2.3 Functional Requirements**

- The software shall provide an authentication system for Jacobs faculty and/or staff member
- The software shall provide an authentication system for external guests
- The software shall recognize QR-Codes which are not released from the admin
- The software shall automatically check-out after 75 minutes
- The software shall store the history of attendance
- The software shall automatically delete the stored data every 14 days
- The software shall show the occupancy of a classroom

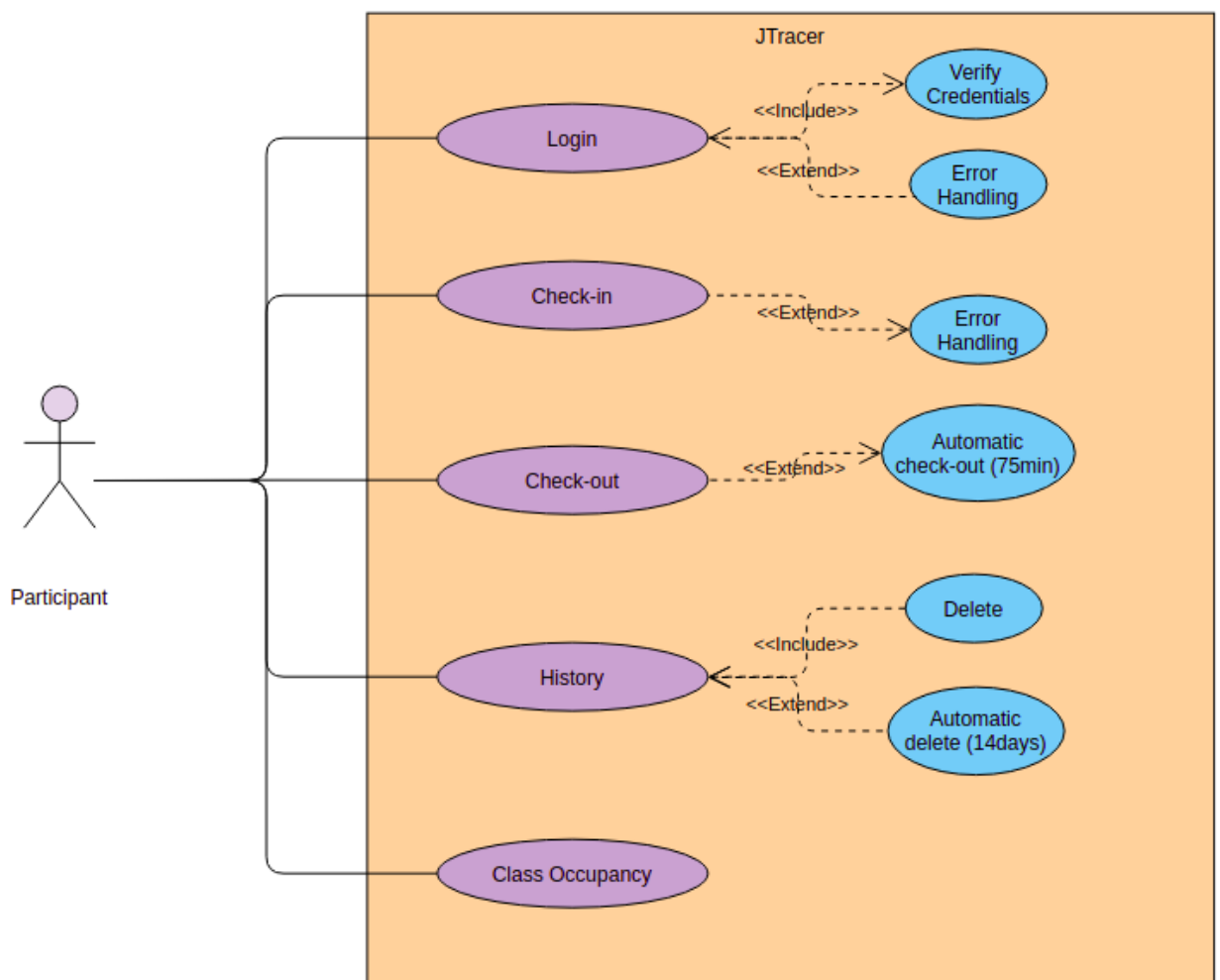
### **2.4 Nonfunctional Requirements**

- The user shall be redirected to home page after successfully checking-in
- The system shall make the number of students in a classroom available to all users

- The application shall be updated upon receiving the status update from JTracer ACS
- Internet connection is required
- The admin should provide the QR-Codes
- The load of the front page does not occupy more than a second maximum for users that access it.
- The response time between click and reaction must average at less than half a second and reach its maximum at 2 seconds.

### 3 System Design

#### 3.1 Use Case Diagram



- **Actors**

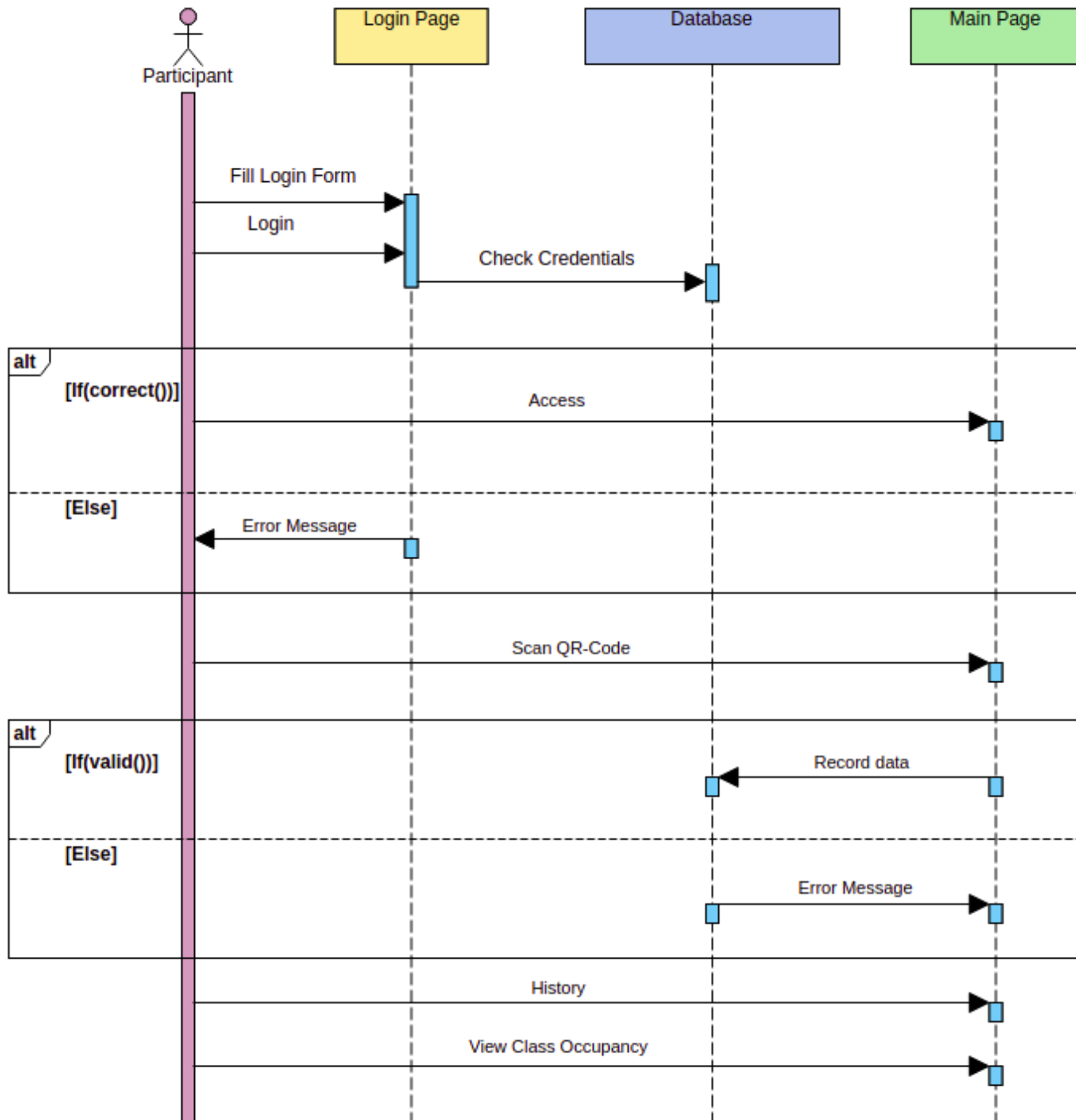
By "Participant" is meant Professor, Student, or any external guest.

- **Roles**

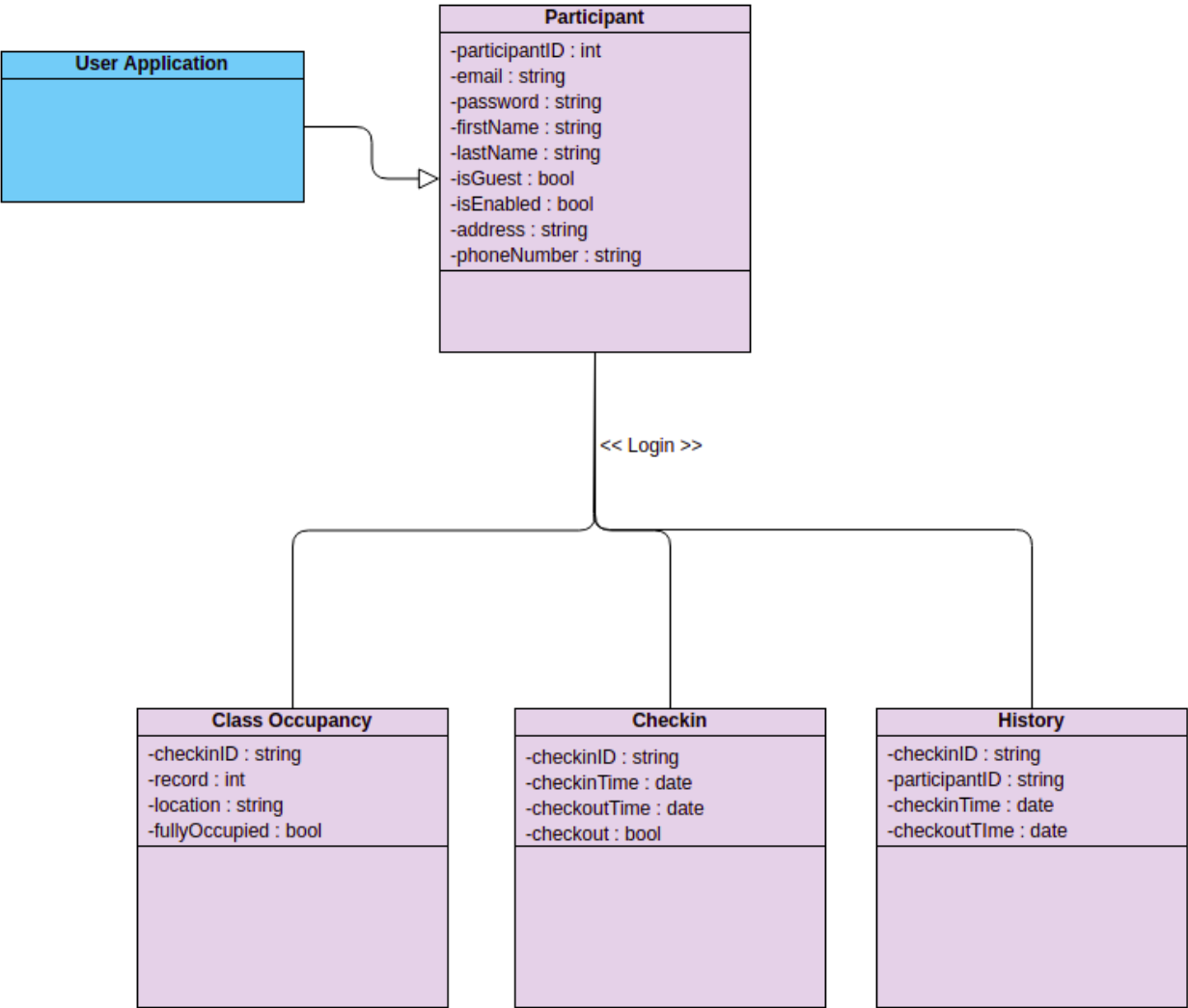
Since in this case any user is considered as a participant, the roles will be the same

relevant to the system.

### 3.2 Sequence Diagram

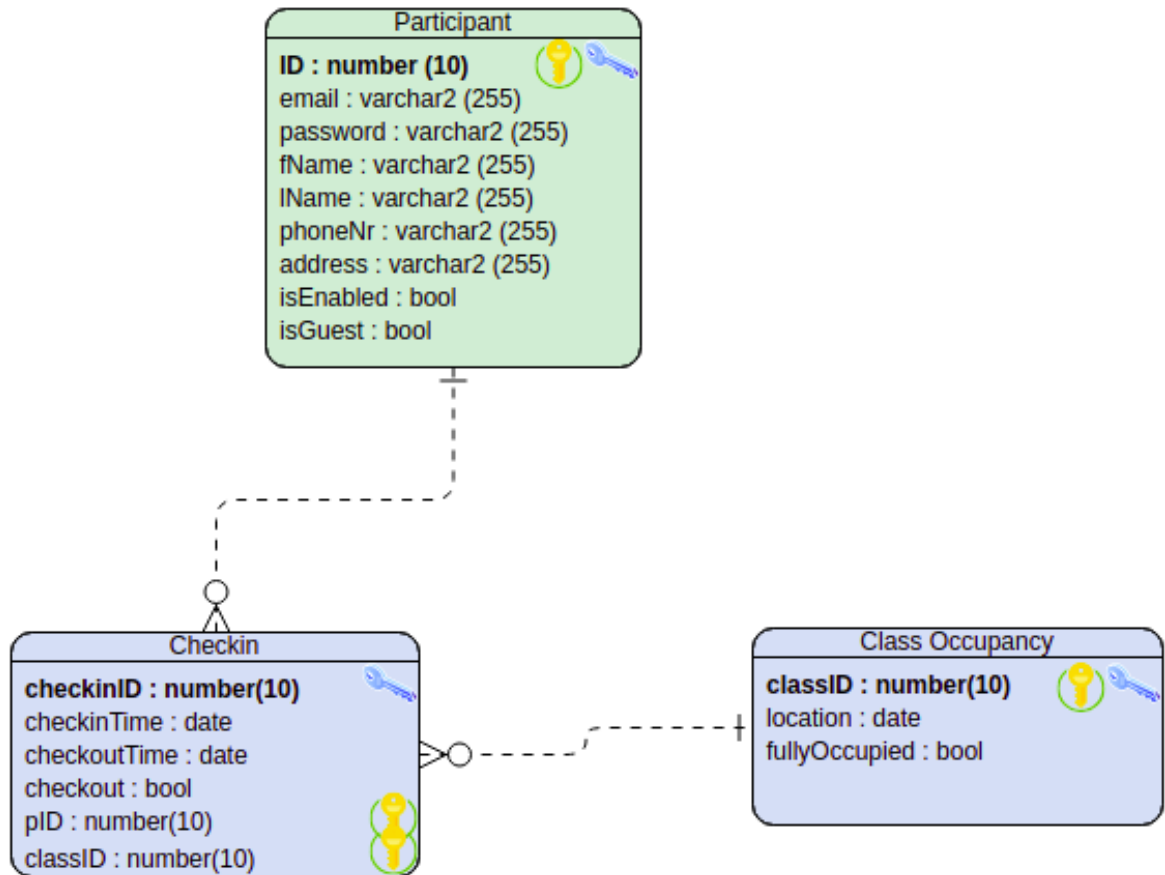


3.3 Class Diagram



## 4 Database Design

### 4.1 ER Diagram



The database design is really simple. As seen from above, there are only three tables, a Participant, Checkin, and Class Occupancy respectively. Each of the tables has a primary key to ensure that the data in the respective column is unique and a foreign key from the Checkin table to Participant and Class Occupancy. The foreign key provides a link between the three tables. Apart from that, I decided to store attributes like *record* and *fullyOccupied* in the software (back-end) in order to reduce the waiting time, and space.

There is a one-to-many relationship between Participant and Checkin as students will be able to checkin in different halls. Furthermore, Checkin is related to Class Occupancy in a many-to-one relationship. That is mainly because of the same reason, many participants can checkin in different classes.

Note: Yellow stands for the foreign key and blue stands for the primary key.