Software requirements specification for project Effective Communication with Dormitory Settlement

1. Authors

- Zhitnik Elizabeth Vladimirovna
- Kozoliy Mikhail Mikhailovich
- Vlasenko Ivan Alexievich
- Kozorez Nikita Fedorovich

Mentor: Mokin Konstantin Yuryevich

2. Introduction

Importance of the project

At the moment, there is no unified system for communication between departments responsible for settling students in dormitories, all work has to be done manually. There is a need to create and implement the necessary system for tracking the residence of students.

Main Concept

We are going to solve the task of creating a component for simplifying the work of tracking the residents in the dormitory. The information is going to be presented in a visual way as a plan of the floor with rooms and occupied places. The data is going to be automatically updated when the status of the rooms changes and provide current information on available places.

3. Glossary

- **Dormitory** – building for students to live in while studying at the university. Students live in groups in rooms, the number of people per room varies.
- **Employee** – person responsible for check-in, check-out, maintaining the database of contracts, control of student accommodation.
- **Dormitory accommodation** – conclusion of a dormitory accommodation agreement with the student, entering it into the database of dormitories.
- **Eviction from the dormitory** – cancellation of the contract with the student for his/her residence in the dormitory.
- **Student Contract** – a specific document with a specific number, validity period. It is necessary to control the student's residence in the hostel, it is concluded when the student moves into the hostel.

4. Actors

4.1 Name: Employee

Description:

The main user of the software who can use all functions of the system.

Main goals:

- student check-in/check-out
- search for students by room number
- room search by student's surname
- processing of notifications about expiry of contracts, student leave, possible violation of accommodation rules

4.2 Name: Software

Description:

All project software necessary for its functioning. Provides a link between the employee and the database of hostels.

Main goals:

- processing of employee's requests
- sending out notifications about expiry of student's contract, student's going on academic leave
- tracking of student contract expiry dates

5. Functional requirements

5.1. Strategic Use-cases

5.1.1. Use-case < Employee's work with the database of dormitories>

Actors: *Employee, Software*

Goals: *Interacting with the student database*

Main success scenario:

1. Employee performs check-in/check-out of students.
2. The system updates the value in the database.
3. The system periodically sends notifications to the employee informing about expiry of the student's contract, student's going on academic leave.
4. The employee processes the notifications.
5. The system updates the database as a result of changes.

5.2. Use-cases for Employee

5.2.1. Use-case < Searching for a student by room >

Requesting information about the student, the room they are registered in, their contract number and term.

Actors: *Employee, Software.*

Goals:

Employee - *to obtain information about the room being searched, the students living in it, their rooms and the term of their contracts. The room being searched, the students living in it, their rooms and contract dates.*

Software – *correct search at the employee's request, interaction with the database.*

Precondition: *Employee selected a dormitory.*

Trigger condition: *A staff member is about to search for a student by room. Clicks the "Search Student by Room" button.*

Main success scenario:

1) Employee submits a request to the system to search for a student by room number.
2) Employee selects a dormitory.
3) The dorm number is correct.
4) The software goes to the list of students living in that dorm.
5) Employee selects a room number.
6) The room number is correct.
7) The software displays information about the students living there.

Alternative scenario < Incorrect dormitory number >:

An employee entered an incorrect dorm number.

1) The software displays the message: "Incorrect dormitory number entered".
2) The software re-requests the dormitory number.
3) The employee re-enters the dormitory number.

Alternative scenario < Incorrect room number >:

The employee entered an incorrect room number.

1) The software displays a message: "Incorrect room number entered".
2) The software re-queries the room number.
3) The employee re-enters the room number.

5.2.2. Use-case < Searching for a room by student >

Requesting information about the student, the room they are registered in, their contract number and term.

Actors: *Employee, Software.*

Goals:

Employee *- get information about the student and the room where he/she lives. Search for the room where the student with this name and information about him/her lives.*

Software *- correct search at the employee's request, interaction with the database.*

Precondition: Employee selected a dormitory**.**

Trigger condition: *A staff member is about to perform a room search by student. The employee clicks the "Search for a room by student" button.*

Main success scenario:

1) An employee submits a request to the system to search for a room by student.
2) The employee enters the student's full name.

3) The student's full name is correct.
4) The software displays information about the student, including the room number where the student is registered.

Alternative scenario < Incorrect student name >:

A staff member entered the student's name incorrectly.

1) The software displays the following message: "Student with this name does not exist".
2) The software re-queries the student's full name.
3) The employee re-enters the query.

<Additional section>:

Notes:

If the student is not registered in any room.

Actors: Software.

Scenario:

The software outputs a message in the information that the student has not checked into any of the rooms.

5.2.3. Use-case < Choosing Dormitory>

The employee selects the dormitory in which further work will be done.

Actors: Employee, Software

Goals:

Employee selected a dormitory, went to the appropriate section.

Main success scenario:

1) The employee is in the dormitory selection section and is about to make a selection.
2) The employee clicks on the appropriate button that presents a specific dormitory.
3) The employee enters the selected dormitory section.

5.2.4. Use-case < Checking a student into a dormitory >

Checking a student into the dormitory, after approval. Entering his/her personal data into the database.

Actors: Employee, Software.

Goals:

The employee entered the student's data, the software updated the database due to the changes.

Precondition: Employee selected a dormitory

Trigger condition: The employee clicked the "Check-in" button

Main success scenario:

1) An employee has entered the student's full name
2) Employee has entered the student's room
3) Employee has entered the student's contract number
4) Employee has entered the student's faculty
5) Employee has entered the student's group number
6) Employee entered remarks related to this student
7) Employee entered the student's statements
8) The software has entered the entered data into the database.

5.2.5. Use-case < Eviction from the dormitory >

Eviction of a student from a residence hall. Deletion of his/her personal data from the database

Actors: Employee, Software.

Goals:

The employee wrote the reason for the selected student's eviction, the software updated the database in consequence of the changes made.

Precondition: Employee selected a dormitory and chose a student.

Trigger condition: The employee selected student with search and clicked the "Evict" button

Main success scenario:

1) The staff member wrote the reason for the eviction.
2) Software deleted information from the database about the evicted student

5.2.6. Use-case < Processing of notifications by employee>

The software periodically sends notifications to the employee, the employee processes them and the software makes changes to the database.

Actors: Employee, Software.

Goals:

Employee saw and processed the notification

Precondition: One of the events took place:

- The contract has expired
- Contract expiration date is approaching
- Student has applied for eviction from the dormitory
- Student has applied to move into the dormitory

The software has sent the appropriate notification

Trigger condition: The employee clicked on the notification of their choice

Main success scenario:

1) The staff member wrote the reason for the eviction.
2) The employee saw the full text of the notice.
3) After completing the required actions, the employee noted that the notification had been processed.
4) The software updated the database according to the changes made

6. Non-functional requirements

6.1. Environment

The platform will be web-based, accessible through modern desktop web browsers:

- Chrome (and Chromium-based)

6.2. Performance

- Sending notifications about changes that have occurred immediately after they occur.
- Correct updating of contract terms.
- Providing results when an employee searches within 1 second or less