

COMPX554-20B

Security System

Part One

Steve Reeves

This document gives the requirements for a security system for doors in a building.

1 A single door

We start with a single door.

1.1 Requirements

A door has its security controlled by a swipe-card system.

The system consists of a database that records for each user of the system what their personal identification number (PIN) is.

The door has two mechanisms, one on the outside (for entering the building) and one on the inside (for leaving the building).

To use the door as an exit, a person simply needs to push the big red button marked “Push To Exit”. The system unlocks the door, allowing the person to leave. The door remains unlocked for a certain amount of time, as given by a system constant. After this time the door is locked.

To enter through the door, the person swipes their card and presses keys that correspond to their four-digit PIN. The system checks that the person has given the correct PIN and that they are allowed through that particular door. If these two conditions are met, the system unlocks the door. The door remains unlocked for a certain amount of time, as given by a system constant. After this time the door is locked. The system also records the fact that the person concerned used the entry mechanism.

The door’s status can be determined at any time by the system, and its status will be exactly one of closed and locked, closed and unlocked or open.

The security system is actually part of a larger building management system (controlling lighting, heating, safety as well as security), so if the fire alarm is activated within the building, the door’s status must be closed and unlocked or open.

1.2 A model

Your task is to build a Z model of the system described in section 1.1.

Your model must make available operations that initialise the system, allow entry, allow exit, unlock a door in an emergency, allow adding a user, deleting a user and amending a user.

You should build the model by having a state schema called *Status* that describes the state of the door at any moment. Your operations change *Status* appropriately (i.e. according to the requirements), but at this level of modelling you should assume the operations are instantaneous, so all the observations of *Status* always have a definite value.

Your model must be presented in the traditional Z way as a sequence of paragraphs, each consisting of some English description of the features of the state or operations or auxiliary machinery (e.g. functions, global constants etc.) you are currently considering, followed by the Z version for that feature. The traditional style is described in the document by John Wordsworth on the Moodle site. Please make sure that you follow it. That document also provides a quick overview of the Z ideas that we have covered, so it is good revision too.

Start with a description of the overall idea of the system, then introduce the state, its initialisation and the operations (in that order), together with any necessary auxiliary definitions. You should consider doing some proofs, for example of an initialisation theorem, and to investigate the pre-conditions of your operations.

Pay attention to making the whole document readable. Check the Z using ProB to make sure it is syntax- and type-correct. The whole document must be written using the L^AT_EX *article* style (just as the one you are currently reading is). Great importance is attached to a high-quality document and a simple, elegant model that meets the requirements.

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

2 More than one door

Once you have built the model as in section 1, use that model of a single door to model a (more realistic) building that has several (some finite number) of such doors.

2.1 Further requirements

The model should be extended so that the system records all the doors in the building, identifying each with a door identification number (DIN), and all the allowable users and their associated PINs. Each DIN should be mapped to a particular door's status (as modelled using a schema very similar to *Status*).

The operations available on a single door should be raised to the system level, so there should be a system initialisation operation that initialises each door, a system-level operation to add a user, a system-level operation to handle a user leaving via a particular door, and so on. This should all be modelled explicitly, *i.e. do not use promotion*, since I want you to write as much Z as possible!

Use ProB to check that your operations have the required effect when tried on a small model. You might find this tricky due to the “explosion” of states, so choosing the various constants (both in the ProB preferences and in your model itself) will be critical.

Make your model as well-structured as you can: a uniform, modular approach that models the requirements in a logical way that reflects the physical structures involved, so as to make your model clear for your client, would be recommended.

Assessment

You should hand in a L^AT_EX file, so please make sure that it compiles (so I can read the PDF version). Your file (model and accompanying, explanatory text) should be handed-in via the Moodle site by 1000 on Monday 28th September 2020. A second part to this coursework will be issued at that time, and it will be due in around three weeks after.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder