

Operation of security devices in a lockout-tagout system

Field

- 5 The present disclosure relates in general terms to operation of security devices in a lockout-tagout system, including, in certain aspects, authentication of users to allow locking or unlocking of the security devices.

Background

10

Industrial or heavy machinery is often dangerous and requires adequate safety procedures in place to ensure the occupational health and safety of the workers in the area. For example, a single piece of machinery may have multiple components that are potentially hazardous such as electricity, hot steam and saw blades.

15

In some industries, multiple trades of workers may require access to a single piece of equipment or worksite at the same time. Whilst the equipment or system is being worked on, it is important for safety purposes that the equipment is not accidentally switched on or functioned. This could result in workers in the vicinity of the equipment or who are working on the equipment or system being injured, or may cause an unwanted occurrence such as pollution or other hazardous or otherwise undesirable events. As such, it is important that the piece of equipment is energised (switched on or functioned) or de-energised (switched off or functioned) only when it is appropriate to do so. This is particularly important for equipment with potential hazardous energy sources where if maintenance is being performed, for example, it is imperative that the machine or the system it belongs to is not reactivated whilst work is being completed.

20 A method of ensuring that a single industrial device is not energised or de-energised (functioned) inadvertently is by using what is known in the industry as a lockout-tagout system.

A Lockout Tagout System (LOTO) is a system in which a mechanical or electrical piece of equipment or system is isolated so that it prevents accidental operation of the item or system. This may be a valve (prevent opening / closing), piece of equipment

35

- 2 -

(prevent from functioning) or electrical isolation to prevent a piece of machinery from being energised.

5 The lockout-tagout system includes a single lockout device which allows one or more padlocks or other locking systems to be placed on the lockout device (i.e. to tag the device) which ensures that only when all the padlocks are removed (i.e. to untag the device), that the equipment can be energised or de-energised (functioned). This is cumbersome because there is often no way for the various parties to know the status of the equipment i.e. is it tagged or ready to be energised (operate), unless they are
10 *physically* present at the locked out/tagged out equipment. Further, the interested party would not know if there is a change in the status of the equipment. In some circumstances, inter-departmental meetings would be held daily or weekly to provide updates on the status of the equipment. In other circumstances, workers would have to go to the equipment in question to determine the status. This is time consuming
15 for the workers resulting in a reduction in worker productivity and efficiency. In addition, the permit system often requires a paper permit to be manually transported to a repository location which is time consuming.

20 In some applications, a permit system may be in place requiring one or more levels of approval from various parties before a system is isolated (LOTO) or re-energised. In some protocols, 3 levels of approval, e.g. isolation/energization permits (signatures required), are required before the lockout device can be isolated or removed. This is a time consuming process where the permit may need to be filled in manually by a supervisor and who will have to receive confirmation from his workers that their
25 department is ready to re-energise the equipment. In some cases, these permits must be executed in sequence, which further delays the entire process and the physical permit has to be handed physically to the next authorising party for their authorisation/approval.

30 It is generally desirable to overcome or ameliorate one or more of the above described difficulties, or to at least provide a useful alternative.

Summary

In accordance with embodiments of the present disclosure, there is provided a method for authenticating a user to change a status of one or more security devices in a lockout-tagout system, each security device being lockable to prevent actuation of a machinery operation mechanism, the method performed by one or more processors in communication with a data store that stores status information for the one or more security devices and user data for a plurality of users of the lockout-tagout system, the one or more processors being configured to perform the steps of:

10

receiving a request to change a status of a security device;
determining, by the one or more processors, if a permit is required to change the status of the security device, and responsive to a determination that a permit is required, performing the following steps:

15

sending, to a computing device of a permit authorizer, a request for a permit level approval;

receiving, from the computing device of the permit authorizer, the permit level approval;

20

determining, by the one or more processors, if the permit is approved at all required permit levels;

issuing, by the one or more processors, a permit approval;

determining, by the one or more processors, if the user is eligible to receive an access token for changing the status of the security device; and

25

responsive to a determination that the user is eligible to receive the access token, sending, to a mobile computing device of the user, the access token, said access token being configured to change the access status of the security device for the user when transmitted to the security device.

In accordance with embodiments of the present disclosure, there is provided a system for authenticating a user to change a status of one or more security devices in a lockout-tagout system comprising one or more security devices being lockable to prevent actuation of a machinery operation mechanism, the system comprising:

one or more processors in communication with:

- 4 -

a data store that stores status information for the one or more security devices and user data for a plurality of users of the lockout-tagout system; and

computer-readable storage having stored thereon instructions which, when executed by the one or more processors, cause the system to:

receive a request to change a status of a security device;

determine if a permit is required to change the status of the security device, and responsive to a determination that a permit is required:

send, to a computing device of a permit authorizer, a request for a permit level approval;

receive, from the computing device of the permit authorizer, the permit level approval;

determine if the permit is approved at all required permit levels;

issue a permit approval;

determine if the user is eligible to receive an access token for changing the status of the security device; and

responsive to a determination that the user is eligible to receive the access token, send, to a mobile computing device of the user, the access token, said access token being configured to change the access status of the security device for the user when transmitted to the security device.

Brief Description of the Drawings

Embodiments of the present disclosure are hereafter described, by way of non-limiting example only, with reference to the accompanying drawings, in which:

Figure 1 is an exemplary lockout-tagout system according to certain embodiments;

Figure 2 is a flowchart of a method for locking a security device;

Figure 3 is a flowchart of a method for adding a tag to a security device;

Figure 4 is a flowchart of a method for removing a tag from a security device;

Figure 5 is a flowchart of a method for approving a permit;

Figure 6 is a flowchart of a method for unlocking a security device;

Figure 7 is a flowchart of a method for an exemplary workflow for de-energising and energising a piece of equipment;

- 5 -

Figure 8 is a flowchart of a method for unlocking a security device showing the interoperation of the components of the system shown in Figure 1;

Figure 9 is a flowchart of a method for unlocking a security device in offline mode showing the interoperation of the components of the system shown in Figure 1;

5 Figures 10 to 16 are exemplary screenshot illustrating, in a mobile format, a mobile application running on a mobile computing device as shown in Figure 1;

Figure 17 is an exemplary screenshot illustrating a main screen running on a computing device as shown in Figure 1;

10 Figure 18 is a block diagram of an example mobile computing device of the system shown in Figure 1; and

Figure 19 is a schematic diagram showing components of an example computing device.

Detailed Description

15

The example system 10 shown in Figure 1 allows secure and efficient management of potentially hazardous equipment. The system 10 comprises the following:

- 20 (a) mobile computing device 12;
- (b) server 16; and
- (c) security device 22.

The components of system 10 are in communication via the network 20. The communication network 20 may include the Internet, telecommunications networks
25 and/or local area networks. In some embodiments, security device 22 is in direct communication with mobile computing device 12, for example by means of Bluetooth or NFC communication.

Security device 22 is a safety lock which can be actuated by electronic means and
30 which can be used to prevent equipment from being operated, typically by interfering with the mechanism for turning the equipment on or off (such as a valve handle, electrical switch or circuit breaker). Exemplary security devices suitable for use with embodiments of the present invention are those manufactured by Master Lock Company LLC of Oak Creek, WI.

35

- 6 -

The system 10 makes managing access to equipment by means of a security device such as a lock simpler, faster and more convenient. Advantageously, system 10 provides for more efficient management of permits, some permits requiring multiple levels of authorisation, for secure authorised access to the equipment.

5

Conventional lockout-tagout systems often require multiple padlocks each with individual tags to ensure that a piece of equipment does not get energised until each of the padlocks and tags are removed. Additionally, multiple permit levels may be required to be authorised before the equipment can be energised. The process is typically inefficient and time consuming. Additionally, it is difficult to monitor the status of each piece of equipment which is undergoing maintenance in large sites such as shipyards. In contrast, embodiments of the present invention provide centralised management of permits and user access tokens such that users are readily able to view the status of each piece of equipment, whether remotely located or not, and to electronically lock, unlock, tag and untag security devices associated with hazardous equipment.

10

15

Locking Process 200

20 An example of a process 200 for locking a security device is shown in Figure 2. At step 202, a request from a user is received, for example from a mobile computing device 12 of the user, to lock a security device 22. The user request may include an application for a new lockout-tagout (LOTO) permit. For example, the request is sent by a mobile application running on mobile computing device 12 to server 16. In other embodiments, the process 200 is performed by mobile computing device 12 in communication with a database 18 such as a cloud database. The server 16 is also in communication with the database 18.

25

In some contexts, it may be necessary to determine if the status of the security device 22 is unlocked. In these contexts, the server 16 receives the request and performs the following steps:

30

- (a) retrieving, from the data store, a status of the security device 22; and
- (b) determining, by the one or more processors, if the status indicates that the security device 22 is unlocked.

35

- 7 -

If the status of the security device 22 is locked, the server 16 may send a notification that the security device 22 is already locked to the user 14 who sent the user request, for example via the mobile application running on mobile computing device 12.

5

At step 204, if the user request to lock a security device 22 includes an application for a new LOTO permit, a permit approval process 500 is performed. Permit approval process 500 is discussed in detail below with reference to Figure 5.

10 Step 206 is then performed, whereby the server 16 determines if the user who sent the request is authorised to lock the security device 22. In some embodiments, this includes receiving, as part of the user request, one or more of the following data elements, and matching the data elements against user data stored in database 18:

- 15
- (a) user ID;
 - (b) mobile device identifier, such as an IMEI, MAC address, or UUID;
 - (c) equipment number; and
 - (d) lock ID.

20 If all permit levels of the permit are authorised, step 210 is performed where a signal is sent from the server 16 to the mobile computing device 12 of the user 14. The signal preferably including an access token, or permit token, which is then transmitted to the security device 22 to actuate the mechanism of the device from unlocked to locked. The security device 22 receives the token and causes the mechanism to be
25 locked. At step 212, the status of the security device 22 is then updated from unlocked to locked. In some embodiments, step 212 includes updating data store 18 that is in communication with server 16. In some embodiments, server 16 is in communication with a cloud database. In some embodiments, step 214 is performed whereby a notification is sent to the user 16, for example via an app running on
30 mobile computing device 12, that the security device is successfully locked.

In some embodiments, the method further includes steps of:

- 35
- (a) retrieving, from the data store 18, user data associated with a group of users; and

- 8 -

- (b) sending, to mobile computing devices 12 of the users in the group of users based on the retrieved user data, a notification that the status of the security device has changed.

5 In some embodiments, the data store 18 is a cloud database.

In an alternative embodiment, the user request received in step 202 is sent from a first user, and the user request is for authorising a second user to change the status of the security device 22 from unlocked to locked. At step 206, the server 16
 10 determines if the second user is authorised to lock the security device 22. In some embodiments, this includes receiving, as part of the user request, one or more of the following data elements, and matching the data elements against second user data stored in database 18:

- 15 (a) user ID associated with the second user;
- (b) mobile device identifier, such as an IMEI, MAC address, or UUID, for a device of the second user;
- (c) equipment number; and
- (d) lock ID.

20

At step 210, the signal is sent from the server 16 to the mobile computing device 12 of the second user. The signal preferably including an access token, or permit token, which is then transmitted to the security device 22 to actuate the mechanism of the device from unlocked to locked. The security device 22 receives the token from the
 25 second user and causes the mechanism to be locked. In some embodiments, step 214 is performed whereby a notification is sent to the first user and/or the second user, for example via an app running on mobile computing device 12, that the security device is successfully locked.

30 ***Adding a Tag to a Security Device Process 300***

Process 300, as particularly shown in Figure 3, is an example of a process for adding a tag to a security device 22. For example, the method includes steps of:

- 9 -

- (a) at step 302, receiving a request from a user 16 to add an active tag to tag data associated with the security device 22;
- (b) at step 306, determining, by the one or more processors, if the user 16 is authorized to add the active tag to the security device 22;
- 5 (d) at step 308, modifying, by the one or more processors, the tag data by adding the active tag, the active tag including data associated with the user; and
- (e) at step 312, sending, to a mobile computing device 12 of the user 16, a notification that the active tag has been added to the security device
- 10 22.

In some embodiments, the process 300 is performed by server 16. In other embodiments, the process 300 is performed by mobile computing device 12 in communication with a database server, such as a cloud database server.

15 The process 300 may further include step 304 to determine if the status of the security device 22 is locked by retrieving from data store 18 the status of the security device 22.

20 More than one active tag may be added to a single security device. For example, each department that needs to work on the equipment that is locked by security device 22 (such as electrical, plumbing, general maintenance, etc.) may add an active tag to prevent the security device 22 from being energised.

25 ***Removing a Tag from a Security Device Process 400***

An example of a process 400 for removing a tag from a security device 22 is shown in Figure 4. For example, the method includes steps of:

- 30 (a) at step 402, receiving a request from a user 16 to remove an active tag from tag data associated with the security device;
- (b) at step 406, determining, by the one or more processors, if the user 16 is authorized to remove the active tag;

- 10 -

- (c) at step 408, determining, by the one or more processors, if the user 16 associated with the request matches data associated with the user who added the active tag;
- (d) at step 410, modifying, by the one or more processors, the tag data by removing the active tag; and
- (e) at step 412, sending, to the mobile computing device 12 of the user 16, a notification that the active tag has been removed from the security device 22.

10 In some embodiments, the process 300 is performed by server 16. In other embodiments, the process 300 is performed by mobile computing device 12 in communication with a database server, such as a cloud database server.

15 The process 400 may further include step 404 to determine if the status of the security device 22 is locked by retrieving from data store the status of the security device 22.

20 In other embodiments, other users with a higher security level may be permitted to remove an active tag even though another user added that active tag, for example a supervisor or manager may have access to do so.

Permit Approval Process 500

25 An example of a process 500 for approving a permit is shown in Figure 5. For example, the permit may be an energization permit for energizing a piece of equipment or a machine. Another example of a permit is a LOTO permit for locking out a particular piece of machinery to prevent others from actuating the machinery. The permit approval process may include one or more permit levels. For example, each permit level may be associated with a type of permit e.g. one for each department - mechanical department, hydraulics department, structural department. Alternatively, each permit level may be associated with a different security level e.g. supervisor, safety officer, project manager. Of course, depending on the type of permit and application of the embodiment, the number of permit levels and/or type of permit level may be changed accordingly. Other types of permits may also be approved as part of process 500. In some embodiments, the permit levels are

- 11 -

approved sequentially i.e. first approved by a supervisor, followed by the safety officer and finally the project manager. However, in other embodiments, the various permit levels may be able to be approved concurrently.

5 For example, the method includes steps of:

- (a) at step 502, receiving a permit approval request from a user 16 ;
- (b) at step 504, sending permit level request to a permit authorizer based on permit authorizer data associated with a permit level requiring approval;
- 10 (c) at step 506, awaiting receipt of permit level approval from the permit authorizer;
- (d) at step 508, determining if there are any other remaining permit levels which require approval based on permit data;
- 15 (e) at step 510, if there are no remaining permit levels which require approval data, sending approved permit data to the user.

For example, the method may also include the steps of:

- 20 (a) receiving a permit approval request associated with a security device 22;
- (b) sending, to a computing device of a permit level authorizer, a permit level approval request;
- (c) receiving, from the computing device of the permit level authorizer, data associated with an approved permit level;
- 25 (d) determining, by one or more processors, if there are remaining permit levels which require approval, and if so, performing steps (b) and (c) until all permit levels which require approval have been approved;
- (e) issuing, by the one or more processors, a permit token;
- 30 (f) modifying, by the one or more processors, permit data based on the permit token;
- (g) sending, to a mobile computing device of the user, a notification that the permit has been approved.

35 In some embodiments, the method above further includes steps of:

- 12 -

- 5
- (h) identifying from the permit data, by the one or more processors, one or more permit tokens of the permit data that have not been issued;
 - (i) sending, to one or more mobile computing devices associated with the one or more users identified in step (h), a notification that a permit token associated with a permit level has been authorized.

10 In some embodiments, the permit approval request may be received from server 16 instead of mobile computing device 12 of the user 14.

10 In some embodiments, after step 502, the method further includes the step of retrieving permit data associated with the request, permit data including the number of permit levels and permit authorizer data associated with each permit level. In other embodiments, the permit approval request includes permit level and permit
15 authorizer data. Permit authorizer data may include a user ID or mobile device ID for sending a permit level request. In some embodiments, this may be by pushing the permit level request to a computing device 12 of the permit authorizer, such as a mobile computing device, via a mobile application for example. In other embodiments, the permit authorizer may receive the permit level request by a messaging service
20 such as an email or text message.

In some embodiments, after step 510, further notifications may be issued to notify a group of users that a particular permit level has been authorized. The method further includes steps of:

- 25
- (a) retrieving, from the data store, user data associated with a group of users; and
 - (b) sending, to mobile computing devices of the users in the group of users based on the retrieved user data, a notification that the status of the security device has changed.

30

In some embodiments, the data store 18 is a cloud database.

Unlocking Process 600

For example, process 600 is a method for authenticating a user 14 to change a status of one or more security devices 22 in a lockout-tagout system, each security device 22 being lockable to prevent actuation of a machinery operation mechanism, the method performed by one or more processors in communication with a data store that stores status information for the one or more security devices and user data for a plurality of users of the lockout-tagout system, the one or more processors being configured to perform the steps of:

10

receiving a request to change a status of a security device 22;
determining, by the one or more processors, if a permit is required to change the status of the security device, and responsive to a determination that a permit is required, performing the following steps:

15

sending, to a computing device of a permit authorizer, a request for a permit level approval;

receiving, from the computing device of the permit authorizer, the permit level approval;

20

determining, by the one or more processors, if the permit is approved at all required permit levels;

issuing, by the one or more processors, a permit approval;

determining, by the one or more processors, if the user 14 is eligible to receive an access token for changing the status of the security device 22; and

25

responsive to a determination that the user 14 is eligible to receive the access token, sending, to a mobile computing device 12 of the user 14, the access token, said access token being configured to change the access status of the security device 22 for the user 14 when transmitted to the security device 22.

In some embodiments, the above-mentioned method further includes the steps of:

30

retrieving, from the data store, a status of the security device 22;

determining, by the one or more processors, if the status indicates that the security device 22 is locked;

retrieving, from the data store, tag data associated with the security device 22;

35

and

- 14 -

determining, by the one or more processors, based on the tag data, whether the security device 22 has any active tags and if so, performing the step of sending a request to one or more users associated with the one or more active tags to remove the active tag.

5

In some embodiments, the user 14 is determined to be eligible to receive the access token when a predetermined number of permit tokens of the permit data have been issued, each permit token being associated with a type of permit

10 An example of a process 600 for unlocking a security device 22 is shown in Figure 6. For example, the method includes a step 602 of receiving a user request to unlock a security device 22. The user request preferably includes user data and an equipment number or security device ID associated with the security device 22. In some embodiments, the request includes an application for a new energization permit.

15

In some embodiments, the process 300 is performed by server 16. In other embodiments, the process 300 is performed by mobile computing device 12 in communication with a database server, such as a cloud database server.

20 In some embodiments, it may be necessary to determine if the status of the security device 22 is locked. In these embodiments, the server 16 retrieves, from the data store 18, a status of the security device to determine, by the one or more processors if the status indicates that the security device is "locked". If the status of the security device is unlocked, the server 16 may send a notification that the security device 22 is
25 already unlocked to the user 14 who sent the user request, for example via mobile application running on mobile computing device 12.

The server 16 then performs a step 604 which includes retrieving the status from data store, or checking if the tag parameter associated with the security device 22 is zero.

30 This step is for checking if there are any outstanding tags (LOTO) on the security device 22 associated with the equipment. If there are outstanding tags on the security device 22, step 606 is performed where a request is sent to the one or more users associated with the one or more outstanding tags to remove their LOTO tags from the security device 22 so that energization may be performed.

35

- 15 -

At step 607, if a permit is required for energization, permit approval process 500 is performed as described above with reference to Figure 5.

5 In step 608 the server 16 determines if the user who sent the request is authorised to unlock the security device 22. In some embodiments, this includes receiving, as part of the user request, one or more of the following data elements, and matching the data elements against user data stored in database 18:

- 10 (a) user ID;
- (b) mobile device ID;
- (c) equipment number; and
- (d) lock ID.

Step 610 is then performed by sending, to a mobile computing device 12 of the user 15 14, the access token (or permit token), said access token being configured to change the access status of the security device 22 from locked to unlocked when transmitted to the security device 22. The access token is then transmitted to the security device 22 to actuate the mechanism of the device from locked to unlocked. The security device 22, upon receipt of the token, actuates the mechanism to unlock it. At step 20 612, the status of the security device 22 is then updated from locked to unlocked. In some embodiments, step 212 includes updating data store of server 16. In some embodiments, server 16 is a cloud database.

At step 614, a notification is sent to the user 16, for example via an app running on 25 mobile computing device 12, that the security device is successfully unlocked and the equipment is ready for energization.

In some embodiments, the method further includes steps of:

- 30 (a) retrieving, from the data store, user data associated with a group of users; and
- (b) sending, to mobile computing devices of the users in the group of users based on the retrieved user data, a notification that the status of the security device has changed.

35

- 16 -

In some embodiments, the data store 18 is a cloud database.

In alternative embodiments, the user request received in step 602 is sent from a first user, and the user request is for authorising a second user to change the status of the security device 22 from locked to unlocked. In this example, the user request further includes, as part of the user request, user ID for identifying the second user. In this embodiment, steps 604, 606 and 607 is as described above. In step 608 the server 16 determines if the second user is authorised to unlock the security device 22. In some embodiments, this includes receiving, as part of the user request, one or more of the following data elements, and matching the data elements against user data stored in database 18:

- (a) user ID associated with the second user;
- (b) mobile device ID for a device of the second user;
- (c) equipment number; and
- (d) lock ID.

Step 610 is then performed by sending, to a mobile computing device 12 of the second user, the access token (or permit token), said access token being configured to change the access status of the security device 22 from locked to unlocked when transmitted to the security device 22. The access token is then transmitted to the security device 22 to actuate the mechanism of the device from locked to unlocked. The security device 22, upon receipt of the token, actuates the mechanism to unlock it. At step 612, the status of the security device 22 is then updated from locked to unlocked. In some embodiments, step 212 includes updating data store of server 16. In some embodiments, server 16 is a cloud database.

At step 614, a notification is sent to the first user and/or the second user, for example via an app running on mobile computing device 12, that the security device is successfully unlocked and the equipment is ready for energization.

Process 1000

An example of a workflow for de-energising a piece of equipment, locking it so that it cannot be energised until all tags and appropriate permits are issued and finally

energising the equipment safely is particularly shown in Figure 8. Of course, modifications to process 1000 are possible. Advantageously, process 1000 is performed by the various components of system 10 as shown in Figure 8.

- 5 At step 1002 of process 1000, all levels of a de-energisation permit are authorised by the relevant users. In some embodiments, this may include more than one level, where each level can only be authorised by a single person, for example a supervisor or manager of a particular department. After the de-energisation permit is fully authorised at all levels, step 1004 is performed, whereby the equipment is de-
10 energised.

At step 1006, all levels of a lock permit are authorised by the relevant users. This enables a security device to be placed on the equipment to ensure that energisation is not possible until the security device is removed. At step 1008, a security device 22 is
15 secured and locked so that the equipment cannot be energised without first unlocking and removing the security device 22. In some embodiments, this step is provided by locking process 200.

At step 1010, the status of the security device 22 i.e. locked, is updated in data
20 storage of server 16 or in the data store 18. At step 1012, an active tag is added to the security device based on a user request to do so. In some embodiments, this step is provided by process 300. At step 1014, an active tag is removed from the security device based on a user request to do so. In some embodiments, this step is provided by process 400.

25 After all active tags have been removed, step 1016 is performed where the unlock permit is authorised by all levels thereby allowing the security device to be unlocked. In some embodiments, this step is provided by process 500. After all levels of the unlock permit is authorised, step 1018 is performed whereby the security device 22 is
30 unlocked. In some embodiments, operations for implementing step 1018 may be those carried out by process 600, 700 or 800. After the security device 22 is unlocked, all levels of an energisation permit are authorised by the relevant users. In some embodiments, this may include more than one level, wherein each level can only be authorised by a single person, for example a supervisor or manager of a particular

- 18 -

department. After the energisation permit is fully authorised at all levels, step 1022 is performed whereby the equipment is energised.

For example, unlocking process 600 includes a method for authenticating a user 14 to
 5 change a status of one or more security devices 22 in a lockout-tagout system 10, each security device 22 being lockable to prevent actuation of a machinery operation mechanism, the method performed by one or more processors configured to perform the steps of:

- 10 (a) receiving a request to change an access status of the user 14 for a security device 22;
- (b) determining, by the one or more processors, if the user 14 is authorized to apply for an access token;
- (c) retrieving, from a data store, permit data associated with the security
 15 device 22;
- (e) determining, by the one or more processors, based on the permit data, whether the user 14 is eligible to receive the access token; and
- (f) sending, to a mobile computing device 12 of the user 16, the access
 20 token, said access token being configured to change the access status of the security device 22 for the user 16 when transmitted to the security device 22.

The status change may be to change the security device status from locked to unlocked or from unlocked to locked as further described below.

25 In some embodiments, the energisation or isolation of a machine may affect other machines or systems which are dependent on the energisation or isolation of the first machine. In these embodiments, the upstream hazard source is isolated first. In other embodiments, both systems may be locked out.

30 ***Unlocking Process 700***

With reference to Figure 8, process 700 is an overall process flow including steps performed by the components in system 10. For example, server 16 may include a
 35 user/communication module 162, a LOTO module 164 and a permit module 166.

Unlocking Process in Offline Mode 800

In some embodiments, the mobile device 12 may not have connection to network 20, e.g. cellular network in certain areas of a worksite may not be available. In these circumstances, process 800 may be carried out. Process 800 is similar to process 700 except the user is pre-approved to receive a user key such as an access token or permit token to change the status of a security device 22 to be used at a later time.

10 Security Device 22

The security device 22 is preferably a wireless padlock being lockable to prevent actuation of a machinery operation mechanism. In some embodiments, the padlock can be in communication with a mobile computing device 12 by wireless means such as Bluetooth technology, NFC, Zigbee technology or Wi-Fi.

In some embodiments, the security device 22 is in communication with the mobile computing device 12, by means of Bluetooth technology for example, and is configured to receive user requests such as a request to change a status of the security device e.g. to lock or unlock. The security device 22 is also configured to provide status updates to the mobile computing device 12 by means of network 20.

In other embodiments, the security device 22 is in communication with server 16, such as via Wi-Fi, for receiving user requests and providing updates to the server on the status of the security device 22.

Mobile Computing Device 12

Figure 18 is a block diagram showing an example mobile computing device 12 in which embodiments of the invention may be practiced. The mobile computing device 12 may be a mobile computing device such as a smart phone, a personal data assistant (**PDA**), a palm-top computer, and multimedia Internet enabled cellular telephones. For ease of description, the mobile computing device 12 is described below, by way of non-limiting example, with reference to a mobile device in the form of an iPhone[™] manufactured by Apple[™], Inc., or one manufactured by LG[™], HTC[™]

and Samsung™, for example.

As shown, the mobile computing device 12 comprises the following components in electronic communication via a bus 1206:

5

- (a) a display 1202;
- (b) non-volatile (non-transitory) memory 1204;
- (c) random access memory ("RAM") 1208;
- (d) N processing components 1210;
- 10 (e) a transceiver component 1212 that comprises N transceivers; and
- (f) user controls 1214.

Although the components depicted in Figure 18 represent physical components, Figure 18 is not intended to be a hardware diagram. Thus, many of the components depicted
15 in Figure 18 may be realized by common constructs or distributed among additional physical components. Moreover, it is certainly contemplated that other existing and yet-to-be developed physical components and architectures may be utilized to implement the functional components described with reference to Figure 18.

20 The display 1202 generally operates to provide a presentation of content to a user, and may be realized by any of a variety of displays (e.g., CRT, LCD, HDMI, micro-projector and OLED displays).

In general, the non-volatile data storage 1204 (also referred to as non-volatile
25 memory) functions to store (e.g., persistently store) data and executable code.

In some embodiments for example, the non-volatile memory 1204 comprises bootloader code, modem software, operating system code, file system code, and code to facilitate the implementation components, known to those of ordinary skill in the
30 art, which are not depicted nor described for simplicity.

In many implementations, the non-volatile memory 1204 is realized by flash memory (e.g., NAND or ONENAND memory), but it is certainly contemplated that other memory types may be utilized as well. Although it may be possible to execute the
35 code from the non-volatile memory 1204, the executable code in the non-volatile

- 21 -

memory 1204 is typically loaded into RAM 1208 and executed by one or more of the N processing components 1210.

5 The N processing components 1210 in connection with RAM 1208 generally operate to execute the instructions stored in non-volatile memory 1204. As one of ordinary skill in the art will appreciate, the N processing components 1210 may comprise a video processor, modem processor, DSP, graphics processing unit (GPU), and other processing components.

10 The transceiver component 1212 comprises N transceiver chains, which may be used for communicating with external devices via wireless networks. Each of the N transceiver chains may represent a transceiver associated with a particular communication scheme. For example, each transceiver may correspond to protocols that are specific to local area networks, cellular networks (e.g., a CDMA network, a GPRS network, a UMTS networks), and other types of communication networks. In
15 some embodiments, transceiver component 1212 includes protocols for Bluetooth or NFC communication.

It should be recognized that Figure 18 is merely an example and in one or more
20 example embodiments, the functions described herein may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over as one or more instructions or code encoded on a non-transitory computer-readable medium 1204. Non-transitory computer-readable medium 204 comprises both computer storage
25 medium and communication medium including any medium that facilitates transfer of a computer program from one place to another. A storage medium may be any available medium that can be accessed by a computer.

Figure 10 shows an exemplary splash screen for logging into a mobile application
30 1218. For example, a user may access an account entering a username and password.

Figure 11 shows an exemplary splash screen for a mobile application 1218 of a user. For example, the user has access to a list of equipment, a LOTO register, an energization list, a permit list, a list of KPIs and project settings.
35

Figure 12 shows an example screenshot of a list of equipment, some equipment having one or more active tags associated with the equipment. A user may easily see whether there are active tags on a particular piece of equipment. Preferably, a user
5 may search for a particular tag number or switch board.

Figure 13 shows an example screenshot of an energization register. For example, a user may search for a particular tag number of switch board. The user may easily determine which of the equipment are energized and which are not.

10

Figures 14 and 15 shows an example permit form associated with particular equipment. For example, a user may access a permit and approve it from the mobile application 1218. Alternatively, the permit may be retrieved by another user. Alternatively, if a permit request is sent to a user, the user may be directed to the
15 example permit as shown on Figures 14 and 15. Of course, depending on the permit, the contents of the permit form may differ.

Figure 16 shows an example of equipment details for a particular equipment according to its tag number. In this example, the equipment is locked out by Lock01. A user
20 may send a request to unlock and energize the equipment by clicking on the icon 1690 as shown on Figure 16. This may in turn start process 600.

Server 16

25 As shown in Figure 19, the server 16 may comprise a computing device. In some embodiments, the server 16 may comprise multiple servers in communication with each other, for example over a local area network or a wide-area network such as the Internet. As described in the preceding section, the server 16 is able to communicate with other components of the system 10 over the wireless communications network
30 20 using standard communication protocols.

The components of the server 16 can be configured in a variety of ways. The components can be implemented entirely by software to be executed on standard computer server hardware, which may comprise one hardware unit or different
35 computer hardware units distributed over various locations, some of which may

- 23 -

require the communications network 20 for communication. A number of the components or parts thereof may also be implemented by application specific integrated circuits (ASICs) or field programmable gate arrays.

- 5 In the example shown in Figure 19, the server 16 is a commercially available server computer system based on a 32 bit or a 64 bit Intel architecture, and the processes and/or methods executed or performed by the server 16 are implemented in the form of programming instructions of one or more software components or modules 1622 stored on non-volatile (e.g., hard disk) computer-readable storage 1624 associated
10 with the server 16. At least parts of the software modules 1622 could alternatively be implemented as one or more dedicated hardware components, such as application-specific integrated circuits (ASICs) and/or field programmable gate arrays (FPGAs).

The server 16 comprises at least one or more of the following standard, commercially
15 available, computer components, all interconnected by a bus 1635:

- (a) random access memory (RAM) 1626;
- (b) at least one computer processor 1628, and
- (c) external computer interfaces 1630:
 - 20 (i) universal serial bus (USB) interfaces 1630a (at least one of which is connected to one or more user-interface devices, such as a keyboard, a pointing device (e.g., a mouse 1632 or touchpad),
 - (ii) a network interface connector (NIC) 1630b which connects the
25 computer system 1600 to a data communications network, such as the wireless communications network 20; and
 - (iii) a display adapter 1630c, which is connected to a display device 1634 such as a liquid-crystal display (LCD) panel device.

30 The server 16 comprises a plurality of standard software modules, including:

- (a) an operating system (OS) 1636 (e.g., Linux or Microsoft Windows);
- (b) web server software 1638 (e.g., Apache, available at <http://www.apache.org>);
- 35 (c) scripting language modules 640 (e.g., personal home page or PHP,

- available at <http://www.php.net>, or Microsoft ASP); and
- (d) structured query language (SQL) modules 1642 (e.g., MySQL, available from <http://www.mysql.com>), which allow data to be stored in and retrieved/accessed from an SQL database 1616.

5

Advantageously, the database 1616 (or data store 1616) forms part of the computer readable data storage 1624. Alternatively, the database 1616 is located remote from the server 16 shown in Figure 19.

- 10 Together, the web server 1638, scripting language 1640, and SQL modules 1642 provide the server 16 with the general ability to allow the other components of the system 10 to communicate with the server 16 and in particular to provide data to and receive data from the database 1616. It will be understood by those skilled in the art that the specific functionality provided by the server 16 to such users is provided by
- 15 scripts accessible by the web server 1638, including the one or more software modules 1622 implementing the processes performed by the server 16, and also any other scripts and supporting data 1644, including markup language (e.g., HTML, XML) scripts, PHP (or ASP), and/or CGI scripts, image files, style sheets, and the like.
- 20 The boundaries between the modules and components in the software modules 1622 are examples only, and alternative embodiments may merge modules or impose an alternative decomposition of functionality of modules. For example, the modules discussed herein may be decomposed into submodules to be executed as multiple computer processes, and, optionally, on multiple computers. For example, as shown
- 25 in Figures 8 and 09 modules may include user/communication module 162, LOTO module 164 and permit module 166. Moreover, alternative embodiments may combine multiple instances of a particular module or submodule. Furthermore, the operations may be combined or the functionality of the operations may be distributed in additional operations in accordance with the invention. Alternatively, such actions
- 30 may be embodied in the structure of circuitry that implements such functionality, such as the micro-code of a complex instruction set computer (CISC), firmware programmed into programmable or erasable/programmable devices, the configuration of a field-programmable gate array (FPGA), the design of a gate array or full-custom application-specific integrated circuit (ASIC), or the like.

35

Each of the blocks of the flow diagrams of the processes of the server 16 may be executed by a module (of software modules 1622) or a portion of a module. The processes may be embodied in a non-transient machine-readable and/or computer-readable medium for configuring a computer system to execute the method. The software modules may be stored within and/or transmitted to a computer system memory to configure the computer system to perform the functions of the module.

The server 16 normally processes information according to a program (a list of internally stored instructions such as a particular application program and/or an operating system) and produces resultant output information via input/output (I/O) devices 1630. A computer process typically comprises an executing (running) program or portion of a program, current program values and state information, and the resources used by the operating system to manage the execution of the process. A parent process may spawn other, child processes to help perform the overall functionality of the parent process. Because the parent process specifically spawns the child processes to perform a portion of the overall functionality of the parent process, the functions performed by child processes (and grandchild processes, etc.) may sometimes be described as being performed by the parent process.

Figure 17 shows an example screenshot of a control panel as displayed to a user. The screen allows the user to view or manage one or more equipment on a vessel, for example.

Throughout this specification, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

The reference to any prior art in this specification is not, and should not be taken as, an acknowledgment or any form of suggestion that the prior art forms part of the common general knowledge.

Claims Defining the Invention

1. A method for authenticating a user to change a status of one or more security devices in a lockout-tagout system, each security device being lockable to prevent actuation of a machinery operation mechanism, the method performed by one or more processors in communication with a data store that stores status information for the one or more security devices and user data for a plurality of users of the lockout-tagout system, the one or more processors being configured to perform the steps of:
- 10 receiving a request to change a status of a security device;
determining, by the one or more processors, if a permit is required to change the status of the security device, and responsive to a determination that a permit is required, performing the following steps:
- 15 sending, to a computing device of a permit authorizer, a request for a permit level approval;
receiving, from the computing device of the permit authorizer, the permit level approval;
determining, by the one or more processors, if the permit is approved at all required permit levels;
- 20 issuing, by the one or more processors, a permit approval;
determining, by the one or more processors, if the user is eligible to receive an access token for changing the status of the security device; and
responsive to a determination that the user is eligible to receive the access token, sending, to a mobile computing device of the user, the access token,
25 said access token being configured to change the access status of the security device for the user when transmitted to the security device.
2. The method of claim 1, wherein the request is a request to lock the security device, and the method further includes steps of:
- 30 retrieving, from the data store, a status of the security device; and
determining, by the one or more processors, if the status indicates that the security device is unlocked.
3. The method of claim 1, wherein the request is a request to unlock a security
35 device, and the method further includes steps of:

- 27 -

retrieving, from the data store, a status of the security device;
determining, by the one or more processors, if the status indicates that the
security device is locked;
retrieving, from the data store, tag data associated with the security device;
5 and
determining, by the one or more processors, based on the tag data, whether
the security device has any active tags and if so, performing the step of
sending a request to one or more users associated with the one or more active
tags to remove the active tag.

10

4. The method of any one of claims 1 to 3, wherein the method further includes
steps of:

retrieving, from the data store, user data associated with a group of users; and
sending, to mobile computing devices of respective users in the group of users
15 based on the retrieved user data, a notification that the status of the security
device has changed.

15

5. The method of any one of claims 1 to 4, wherein the method includes steps of:
receiving a request from a user to add an active tag to tag data associated with
20 the security device;
determining, by the one or more processors, if the user is authorized to add
the active tag to the security device;
modifying, by the one or more processors, the tag data by adding the active
tag, the active tag including data associated with the user; and
25 sending, to a mobile computing device of the user, a notification that the active
tag has been added to the security device.

20

25

6. The method of claim 5, wherein the method includes steps of:
receiving a request from a user to remove an active tag from tag data
30 associated with the security device;
determining, by the one or more processors, if the user is authorized to
remove the active tag;
determining, by the one or more processors, if the user associated with the
request matches data associated with the user who added the active tag;

30

- 28 -

modifying, by the one or more processors, the tag data by removing the active tag; and
sending, to the mobile computing device of the user, a notification that the active tag has been removed from the security device.

5

7. The method of any one of claims 1 to 6, wherein the user is determined to be eligible to receive the access token when a predetermined number of permit tokens of the permit data have been issued, each permit token being associated with a type of permit.

10

8. The method of any one of claims 1 to 7, wherein the method includes steps of:

- (a) receiving a permit approval request associated with the security device;
- (b) sending, to a computing device of a permit level authorizer, a permit level approval request;
- 15 (c) receiving, from the computing device of the permit level authorizer, data associated with an approved permit level;
- (d) determining, by the one or more processors, if there are remaining permit levels which require approval, and if so, performing steps (b) and (c) until all permit levels which require approval have been
- 20 approved;
- (e) issuing, by the one or more processors, a permit token;
- (f) modifying, by the one or more processors, permit data based on the permit token;
- (g) sending, to a mobile computing device of the user, a notification that
- 25 the permit has been approved.

9. The method of claim 8, wherein the method further includes steps of:

- (h) identifying from the permit data, by the one or more processors, one or more permit tokens of the permit data that have not been issued;
- 30 (i) sending, to one or more mobile computing devices associated with the one or more users identified in step (h), a notification that a permit token associated with a permit level has been authorized.

10. A system for authenticating a user to change a status of one or more security
35 devices in a lockout-tagout system comprising one or more security devices being

lockable to prevent actuation of a machinery operation mechanism, the system comprising:

one or more processors in communication with:

5 a data store that stores status information for the one or more security devices and user data for a plurality of users of the lockout-tagout system; and

computer-readable storage having stored thereon instructions which, when executed by the one or more processors, cause the one or more processors to:

10 receive a request to change a status of a security device;
determine if a permit is required to change the status of the security device, and responsive to a determination that a permit is required:

15 send, to a computing device of a permit authorizer, a request for a permit level approval;

receive, from the computing device of the permit authorizer, the permit level approval;

determine if the permit is approved at all required permit levels;

20 issue a permit approval;

determine if the user is eligible to receive an access token for changing the status of the security device; and

25 responsive to a determination that the user is eligible to receive the access token, send, to a mobile computing device of the user, the access token, said access token being configured to change the access status of the security device for the user when transmitted to the security device.

11. The system of claim 10, wherein the request is a request to lock the security device, and the instructions further comprise instructions which, when executed by the
30 one or more processors, cause the one or more processors to:

retrieve, from the data store, a status of the security device; and

determine if the status indicates that the security device is unlocked.

- 30 -

12. The system of claim 10, wherein the request is a request to unlock a security device, and the instructions further comprise instructions which, when executed by the one or more processors, cause the one or more processors to:

retrieve, from the data store, a status of the security device;

5 determine if the status indicates that the security device is locked;

retrieve, from the data store, tag data associated with the security device; and
determine, based on the tag data, whether the security device has any active tags and if so, send a request to one or more users associated with the one or more active tags to remove the active tag.

10

13. The system of any one of claims 10 to 12, wherein the instructions further comprise instructions which, when executed by the one or more processors, cause the one or more processors to:

retrieve, from the data store, user data associated with a group of users; and

15 send, to mobile computing devices of respective users in the group of users based on the retrieved user data, a notification that the status of the security device has changed.

14. The system of any one of claims 10 to 13, wherein the instructions further
20 comprise instructions which, when executed by the one or more processors, cause the one or more processors to:

receive a request from a user to add an active tag to tag data associated with the security device;

determine if the user is authorized to add the active tag to the security device;

25 modify the tag data by adding the active tag, the active tag including data associated with the user; and

send, to a mobile computing device of the user, a notification that the active tag has been added to the security device.

30 15. The system of claim 14, wherein the instructions further comprise instructions which, when executed by the one or more processors, cause the one or more processors to:

receive a request from a user to remove an active tag from tag data associated with the security device;

35 determine if the user is authorized to remove the active tag;

- 31 -

determine if the user associated with the request matches data associated with the user who added the active tag;
modify the tag data by removing the active tag; and
send, to the mobile computing device of the user, a notification that the active tag has been removed from the security device.

16. The system of any one of claims 10 to 15, wherein the user is determined to be eligible to receive the access token when a predetermined number of permit tokens of the permit data have been issued, each permit token being associated with a type of permit.

17. The system of any one of claims 10 to 16, wherein the instructions further comprise instructions which, when executed by the one or more processors, cause the one or more processors to:

- (a) receive a permit approval request associated with the security device;
- (b) send, to a computing device of a permit level authorizer, a permit level approval request;
- (c) receive, from the computing device of the permit level authorizer, data associated with an approved permit level;
- (d) determine if there are remaining permit levels which require approval, and if so, performing steps (b) and (c) until all permit levels which require approval have been approved;
- (e) issue a permit token;
- (f) modify permit data based on the permit token;
- (g) send, to a mobile computing device of the user, a notification that the permit has been approved.

18. The system of claim 17, wherein the instructions further comprise instructions which, when executed by the one or more processors, cause the one or more processors to:

- (h) identify, from the permit data, one or more permit tokens of the permit data that have not been issued;
- (i) send, to one or more mobile computing devices associated with the one or more users identified in step (h), a notification that a permit token associated with a permit level has been authorized.