Volvo CoPilot Safe Assist
Group 3
Design Description

November 30, 2017

Contents

1 Introduction

1.1 Background

Volvo Construction Equipment is a company that develops, manufactures and markets equipments for construction and related industries. One of the problems that they have encountered in their construction sites is the high risk of accidents due to the close proximity of the worker with the machines at work.

As an assignment in the Software Engineering 2 course at Mälardalens University, Björn Brattberg, acting as a representative of Volvo CE, has contracted the Optimus Octavian group to create an application which will reduce the number of accidents at construction sites. This application, called Safe Assist, will run both on the workers' handheld devices and the machines' CoPilot - a display which provides real-time intelligence to the operator working on a machine. It will be Android-based and will issue alarms to both machine and worker, which is not the driver of the machine, whenever there is close proximity between the former and the latter. More importantly however, it will give the operator of the said machine a warning that there are unprotected people close by.

1.2 Definitions

Co-Pilot - a display used in Volvo's construction vehicles. It is an Android-based tablet with applications that have the ability to provide the driver of the vehicle with real-time intelligence from the vehicle. Custom applications that gather information such as the speed of the vehicle, the direction of the movement, the load size, fuel consumption are run in it. Currently, the Co-Pilot offers applications like Dig Assist, Load Assist and Pave Assist, to name some.

2 System Description

2.1 Use Case Diagram

The use case diagram of the Safe Assist application is displayed in Figure 1.

2.1.1 Actors

- Worker: The worker's device with the Safe Assist application installed on it. Can login into the Safe Assist System, associate operator, receive alarm and notification.
- Machine-Operator: The Co-Pilot device inside the machines with the Safe Assist application installed on it. Receives alarm and notification.
- Manager: Supervisor of the Safe Assist in the construction site. Sets the construction site area, can edit and delete it. Can also add, edit and delete users.
- Administrator: Administrator of the users. Can add, edit and delete users.

2.1.2 Use Case Description

- Use Case 1

  ->**Name:** Login

  ->**Initiator:** Worker

  ->**Goal:** Enter into the system to use the services offered by the application

  ->**Main Scenario:**

  1. The Worker enters data to login.
  2. The system checks if the given info for the Worker is valid
  3. System allows the user to get into the system.

  ->**Extensions:**

  1. Given info not valid
     a) System throws an ERROR message
     b) Resume at 1

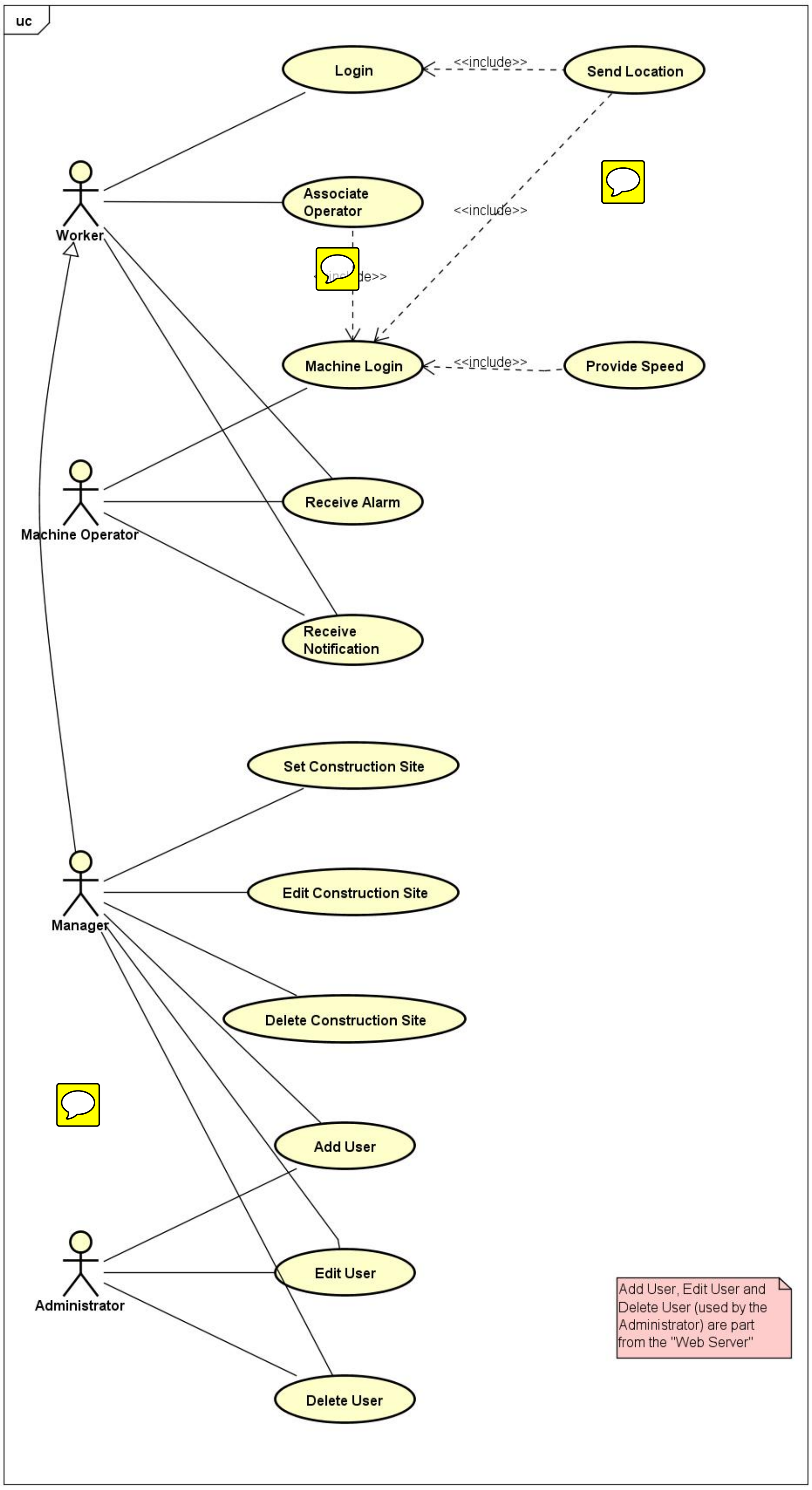


Figure 1: Use case diagram of Safe Assist

- Use Case 2

  ->**Name:** Send Location

  ->**Initiator:** Worker, Machine-Operator

  ->**Goal:** For the installed application to recieve and send the location of the device upon the login.

  ->**Main Scenario:**

  1. The device try to send its location to the server

  ->**Extensions:**

  1. Not possible to establish connection with the device
     a) System throws an ERROR message
     b) Resume at 1

- Use Case 3

  ->**Name:** Associate Operator with Vehicle

  ->**Initiator:** Worker

  ->**Goal:** For the worker to associate himself with a vehicle when using it

  ->**Main Scenario:**

  1. Worker is inlogged in the handheld device
  2. Worker turns on the vehicle and the Co-Pilot Safe Assist
  3. Worker associates himself into the Co-Pilot Safe Assist

  ->**Extensions:**

  3 The association between the operator and vehicle failed
     a) System throws an ERROR message
     b) Resume at 3

- Use Case 4

  ->**Name:** Machine Login

  ->**Initiator:** Machine-Operator

  ->**Goal:** Enter into the system to use the services offered by the application

  ->**Main Scenario:**

  1. The Machine-Operator starts the application.
  2. The Co-pilot application tries to log in to the operator.

  ->**Extensions:**

  2 The application failed to log the user in.
     a) Resume at 2

- Use Case 5

  ->**Name:** Provide Speed

  ->**Initiator:** Machine-Operator

  ->**Goal:** Provide speed to the server to let it to check the Machine-Operators position

  ->**Main Scenario:**

  1. The application requests for the speed of vehicle
  2. The vehicle provides its speed
  3. The device sends the speed to the server

  ->**Extensions:**

3 Not possible to get the machine speed with the device
   a) Resume at 2

- Use Case 6

  ->**Name:** Receive Alarm

  ->**Initiator:** Worker, Machine-Operator

  ->**Goal:** Receive alarm on both devices when a close proximity between the machine and the worker is detected

  ->**Main Scenario:**

  1. The handheld device or Co-Pilot device is inlogged.
  2. An alarm is issued to any of those connected devices when close proximity is calculated.
  3. The devices get the alarm and show it on the screen.

  ->**Extensions:**

  2 There are no provided information about the connected devices.
     a) The system waits for devices to connect
     b) Resume at 1

- Use Case 7

  ->**Name:** Receive Notification

  ->**Initiator:** Worker, Machine-Operation

  ->**Goal:** Receive notification when the device gets inside the Construction Site

  ->**Main Scenario:**

  1. The user is has logged in and connected
  2. The user receives a notification just after she/he gets into the Construction Site

  ->**Extensions:**

  1 The users connection is lost.
     a) The system waits for devices to connect
     b) Resume at 1

- Use Case 8

  ->**Name:** Set Construction Site

  ->**Initiator:** Manager

  ->**Goal:** For the Manager to set a new construction site

  ->**Main Scenario:**

  1. The manager has logged in.
  2. The administrator chooses to set a new construction site.
  3. Location and the scale of the site are inserted and set by the manager to the system.

  ->**Extensions:**

  3 The information about the site is invalid.
     a) System throws an ERROR message
     b) Resume at 2

- Use Case 9

  ->**Name:** Edit Construction Site

  ->**Initiator:** Manager

  ->**Goal:** For the Manager to edit an existing construction site

  ->**Main Scenario:**

1. The manager has logged in.
2. The manager picks the existing Construction Site to edit.
3. The new values of the site are inserted by the manager.

->**Extensions:**

3 The information about the site is invalid.
a) System throws an ERROR message
b) Resume at 2

- Use Case 10

  ->**Name:** Delete Construction Site

  ->**Initiator:** Manager

  ->**Goal:** For the Manager to delete an existing construction site

  ->**Main Scenario:**

  1. The manager has logged in.
  2. The manager chooses to delete an existing construction site.
  3. The manager chooses the site and deletes it.

  ->**Extensions:**

  3 The existing site could not be deleted.
  a) System throws an ERROR message
  b) Resume at 3

- Use Case 11

  ->**Name:** Add User

  ->**Initiator:** Manager, Administrator

  ->**Goal:** For the manager or administrator to add a new user into the system.

  ->**Main Scenario:**

  1. The manager/administrator has logged in.
  2. The manager/administrator chooses to add a new user.
  3. Information about the new user is inserted and set by the manager/administrator to the system.

  ->**Extensions:**

  3 The information about the user is invalid.
  a) System throws an ERROR message
  b) Resume at 3

- Use Case 12

  ->**Name:** Edit User

  ->**Initiator:** Manager, Administrator

  ->**Goal:** For the manager or administrator to edit an existing user.

  ->**Main Scenario:**

  1. The manager/administrator has logged in.
  2. The manager chooses to edit an existing user.
  3. The new values of the site are inserted by the manager/administrator.

  ->**Extensions:**

3 The information about the user is invalid.
a) System throws an ERROR message
b) Resume at 3

- Use Case 13

  ->**Name:** Delete User

  ->**Initiator:** Manager, Administrator

  ->**Goal:** For the manager or administrator to delete an existing user

  ->**Main Scenario:**

  1. The manager/administrator has logged in.
  2. The manager chooses to delete an existing use.
  3. The manager chooses the user and deletes it.

  ->**Extensions:**

  3 The existing user could not be deleted.
  a) System throws an ERROR message
  b) Resume at 3

2.2 High-level Description

The Safe Assist system will be an Android application that will run in machine's CoPilot and in the handheld devices (Android smart phones) of the operators. The application will make use of the combined positioning information provided by operators' handheld devices, and the positioning information provided by the construction machines. When an operator that is not the driver of a certain machine is approaching that machine, the operator's handheld device, as well as the machine's CoPilot, will issue alarms. This will occur if both machine and operator are in a construction area.

In our design, we have 4 actors (Worker, Machine, Administrator and Manager). The worker will provide his location while he is inside a construction area for the system to determine if he is too close to a construction machine. In this case, the worker will receive an issue alarm. He will also receive a notification when he enters to a construction area.

The machine will provide its location to the application to know if it is close to a worker and there is risk of an accident. In case of risk, the machine will receive an alarm. The machine will also provide its speed to amplify the security distance whenever it's moving. Alike the worker, the machine has to login in to the application. The manager will have the same features that a worker, but he will also be able to edit, delete or set a construction area (place where the workers and the manager will provide his their location). The administrator will be able to add, delete or edit users to the database in addition.

2.3 System Overview

As mentioned above, the Safe Assist will provide the operator/s and the machine with alerts whenever close proximity between the two is calculated. To do such, he must be provided with the geographical position of the two units (operator and machine) within the construction site.

The Safe Assist, running in the operator's handheld device, will receive his/her geographical position by getting continuous updates from the handheld device's GPS (which will run in the background, together with the application). On each location input from the operator, the Safe Assist will update his/her coordinates on the database and will update the view of the GUI according to the proper calculations.

The Safe Assist, running in the machine's CoPilot, will gather the information needed by taking input from the CoPilot. The CoPilot, being itself an existing hardware, not to be developed in this project, will communicate with the Safe Assist through a framework, which the Safe Assist will be adapted to. In order to know the geographical position of the vehicle, the GPS system of the CoPilot will run in the background, together with the Safe Assist and updates of the position of the vehicle will be provided to the Safe Assist. In order to get the speed of the vehicle, the Safe Assist will communicate with the hardware - CoPilot - and will take inputs from it to gather this specific information. On each information input (being it vehicle speed and geographical coordinates), the proper updates will be done in the database and the GUI of the Safe Assist will be changed according to proper calculations.

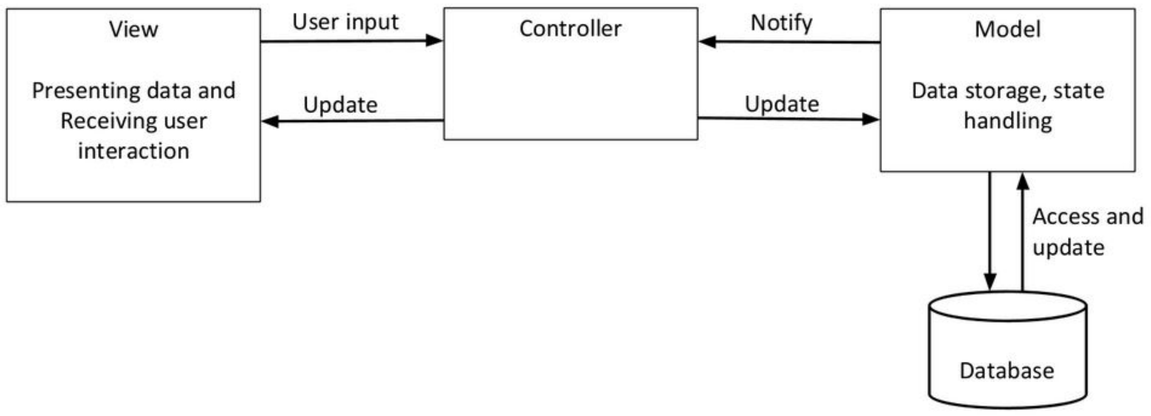

View
Presenting data and Receiving user interaction
User input
Update
Controller
Notify
Update
Model
Data storage, state handling
Access and update
Database

Figure 2: MVC Model of Safe Assist

3 Software Architecture

3.1 Overview and Rationale

The systems overall software architecture is a Model-View-Controller architecture (MVC) based on the inherent structure of an Android application. As an extension of the model part of the architecture, a remote database and web server is used for shared data and exhausting computations.

The Android application will have separate components: server communication, location, unit (device), user interface and controller. The interaction between the parts will be based on MVC interaction as shown in Figure 2. The controller is the hub, or director, in our system. That component receives user inputs from the user interface. Moreover, it requests data and updates the database via the server communication component. The web server acts as an interface for the application to request and receive data from the database. It also works as an interface to administrate the database, e.g. add or remove operators.

One of the main reasons to use MVC is that the application will be used on devices with differing screen sizes. Having a separate component for the user interface allows for an effortless adaptation to the various conditions. A centralized data storage and computation aiding server were chosen due to the need for shared data and to unload the battery-powered mobile devices with heavy workloads.

3.2 System States

State diagram of the Safe Assist is displayed in Figure 3. When Safe Assist application is **initialized**, it identifies the unit and it enters the **Login** state depending on which device it is activated upon. In case it is activated in the CoPilot, after logging in, is goes into **Set Operator** state, which associates the operator, who will run the vehicle, with the vehicle. Afterwards, if successful, it enters the state of **Outside Construction Site**. On a regular interval, it checks if it is inside the construction area and if that is true, it goes into the state of **Inside Construction Area**, and on entering the state it issues a notification to the unit. This state is checked by the continous event of sending Coordinates. After the appropriate calculations, there are two possible states: **Risk level 1** - which issues alert messages - and **Risk level 2** - which issues alarm. In the case of No GPS or Internet Connection, the Safe Assist passes on the **No Connection** state which issues warnings and establishes GPS and Internet Connection.

3.3 Hardware/Software Mapping

The Safe Assist application will run in two devices: the workers' handheld devices and the machines' CoPilot. The database will run in a server. A web interface will run on the computer of the manager and/or the administrator to allow them to request and receive data from the database. Communication through the hardwares will be done through Internet connection and GPS.

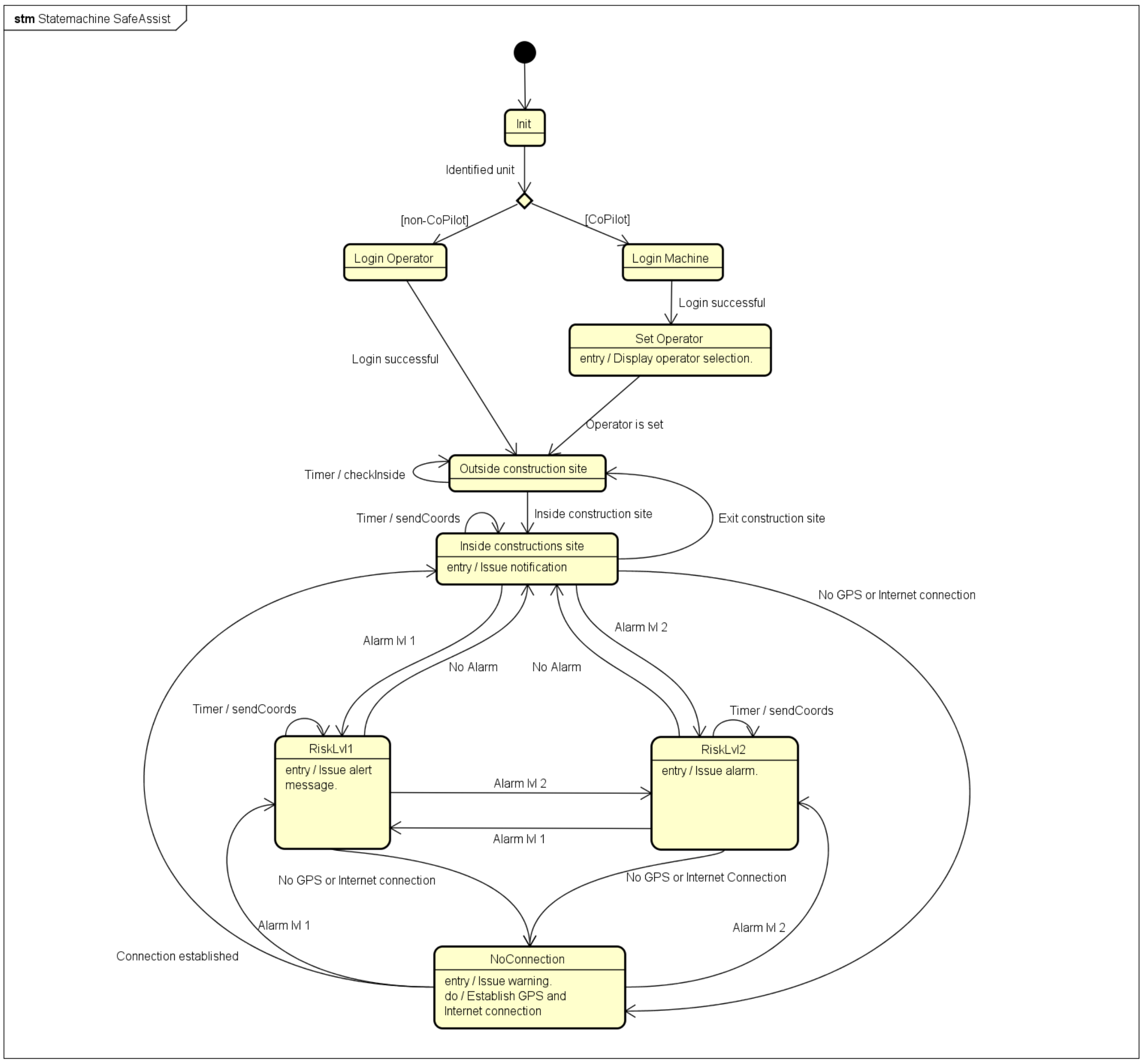


Figure 3: State Diagram of Safe Assist

3.4 Persistent Data

Persistent data in the Safe Assist application is comprised of data in the database which provides the manager and administrator rights to specific people within the company. Location data of workers outside the construction site will not be stored in the database.

3.5 Access Control

This product (Safe Assist application) will be used exclusively by Volvo CE equipment. ID and password authentication method will be used in the application to grant its users access to it.

3.6 Synchronization and timing

The exact techniques and intervals are yet to be decided on due to that no specifics where required or proposed by the client. Further investigations are to be conducted in the matter.

3.7 Start-Up and Shut-Down

The start-up process includes the process of identifying the device where the Safe Assist application has been booted up on.

3.8 Error Handling

In the situation where Internet connection and/or GPS signal is lost, a notification will be displayed in the user interface and the application will try to reestablish the connection. Further investigations are to be conducted in the matter.

4 Software Design Description

4.1 System Components

- Client
  - Component Interface Description

    The interface of the client will be different depending on the privileges of the actor using it. If it is a basic account (the user is a worker or a machine operator), he will have two options: to exit the application and run it on background or not. If he does not exit the application, there will be nothing but a colored box popping up with a short message depending on the level of the alert. If the worker or the machine operator chooses to exit the application and run it on background, the alerts will pop up as a bubble (message) in the handheld device.

    If the account belongs to a manager there will be a top-down view of the work site with dots representing the other units and their location, and also the possibility to add construction area, edit construction area and delete construction area. The manager can be worker or machine operator also, and be physically on the construction area. The application interfaces for the manager will be the same as the worker/machine operator, by adding the extra functionalities of adding/editing/deleting construction areas. All the data such as the user(worker / machine operator) location will come from the web server, and comparing the distance between users, we will be able to provide alerts on different risk levels. Any output will come in the form of data going to the Model of our MVC-structure.

    If the user is administrator he is going to have a web page application, where he will manage all the workers and machine operators. The manager can add operators (workers and machine operators), edit and delete them. On the web page a list of all workers and machine operators categorized by the construction area Id is available.

    A user(worker, machine operator or manager) is going to be identified by a unique personal Id and the construction area Id where he will be working.
  - Component Design

    As seen in the class diagram for the Client (depicted in Figure 4), there are seven classes. Controller, Geolocation, Servercom, Alarm data, Unit, Handheld device and Co-Pilot. The

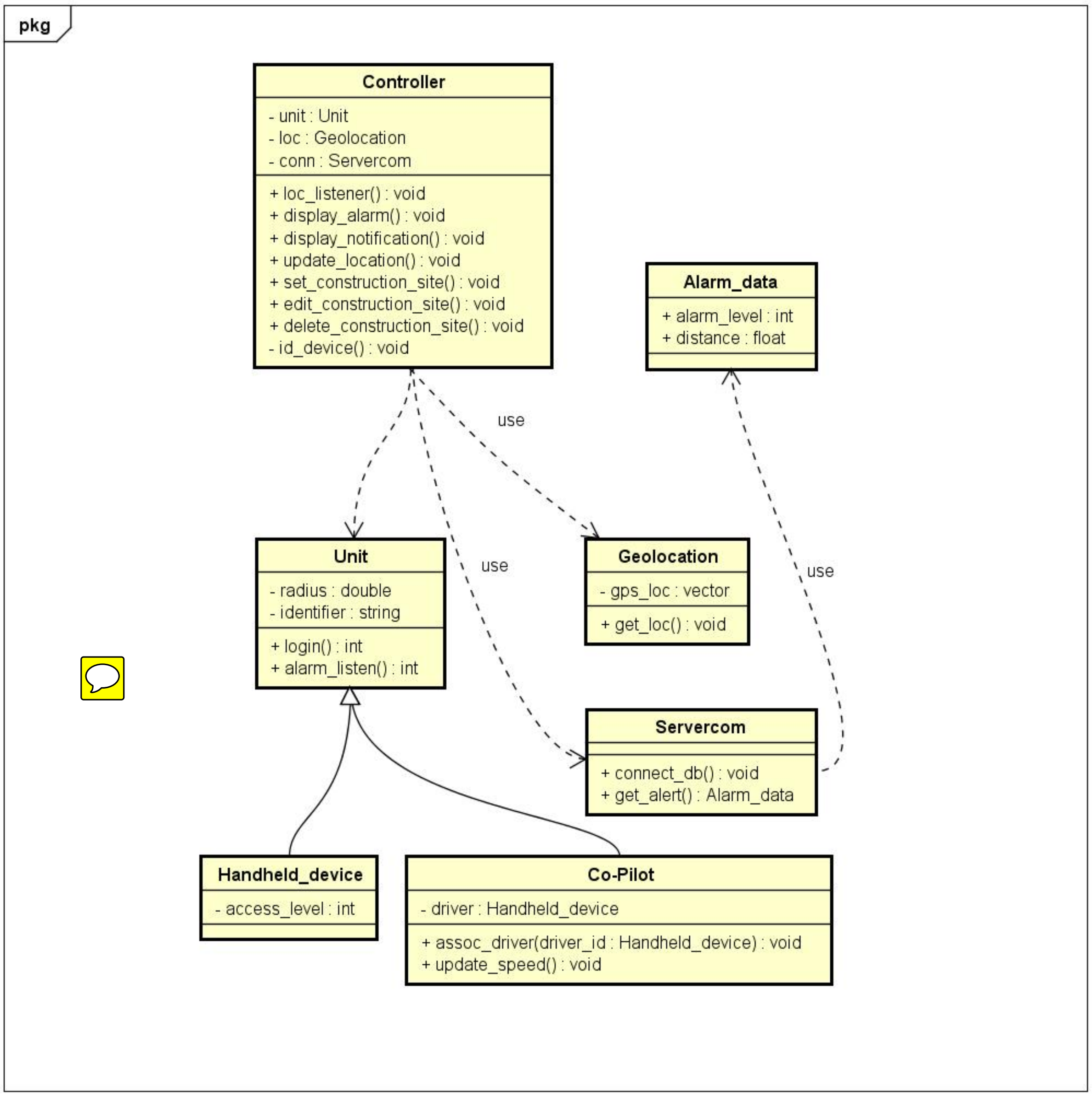


Figure 4: Class Diagram of the Client

class Controller is a standard base-class in Android that serves as an actual controller for everything that goes on in the application. For the purposes of our particular application Controller will contain an object each of the classes Unit, Geolocation and Servercom. It will have methods to allow communication with the server as well to present information on the actual device. By having an object of the class Geolocation, the controller will have access to the location-data native to its own device as well as a means to communicate with the server for any and all needed location-data. The Unit-member provides the identity of the user and the type, be it a handheld device or a Co-Pilot. Through the Servercom-object it will get data about any and all alarms or notifications and display these in the app as required.

Unit is considered a superclass, or as diversely-called: parent-class, to the classes Handheld-device and Co-Pilot. It provides its inheritors with a radius and login-functionality. The inheritors themselves (Handheld device and Co-Pilot) are outfitted with members and methods with intentionally self-descriptive names.

– Workflows and Algorithms

  Depicted in Figure 4.

- Web Server

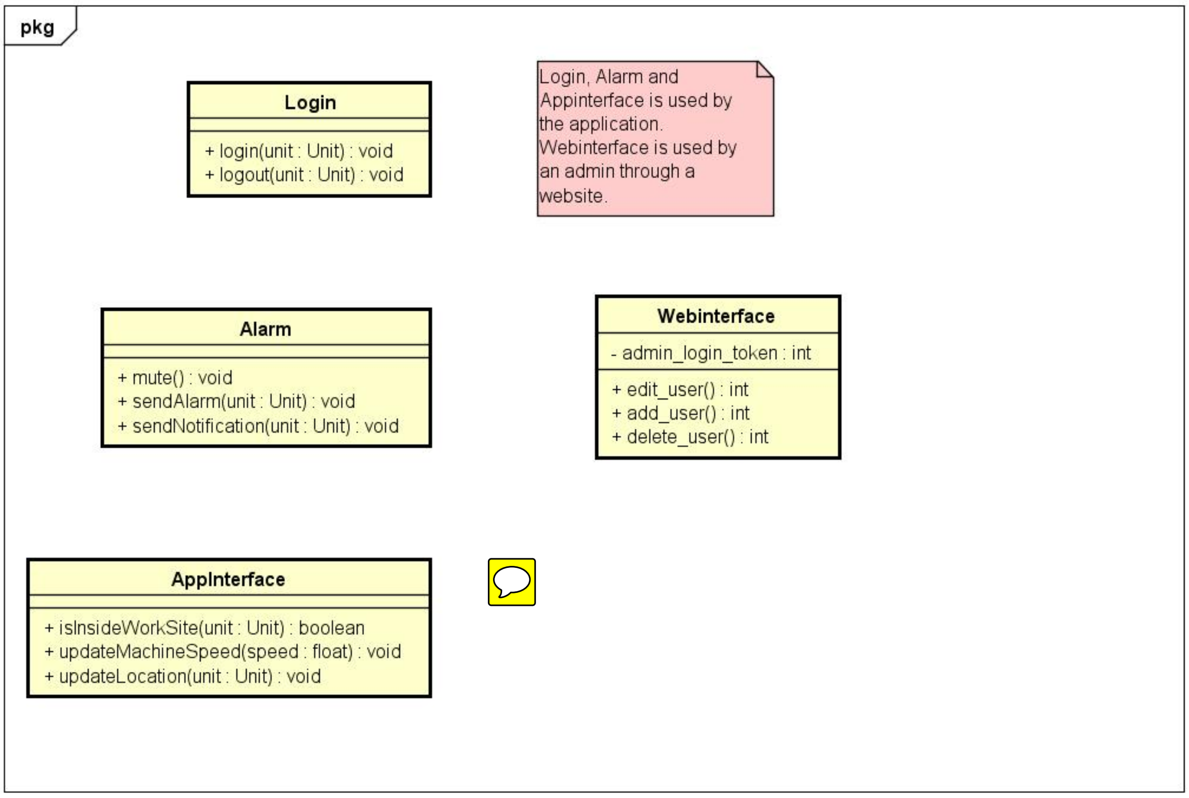

pkg
Login
+ login(unit : Unit) : void
+ logout(unit : Unit) : void
Login, Alarm and Appinterface is used by the application. Webinterface is used by an admin through a website.
Alarm
+ mute() : void
+ sendAlarm(unit : Unit) : void
+ sendNotification(unit : Unit) : void
Webinterface
- admin_login_token : int
+ edit_user() : int
+ add_user() : int
+ delete_user() : int
AppInterface
+ isInsideWorkSite(unit : Unit) : boolean
+ updateMachineSpeed(speed : float) : void
+ updateLocation(unit : Unit) : void

Figure 5: Class Diagram of the Server

- Component Interface Description As a user starts the application. the client (Safe Assist application) will set up communication with the server to check login information at first. Once the login is confirmed the server will use the location-data being sent to it, to see if the client is inside a working area, update any potential speed parameters and update location-data. All the data being sent (locations, speeds, areas and radius) is then used to send appropriate alerts, e.g. entering-notification or the level of proximity-alert. The web interface is solely used by an administrator to manage user-accounts.
- Component Design

  Initial class diagram of the Server component is displayed in Figure 5.

  There are four classes as you can see in class-diagram of the server (Figure 5). The Login-class handles any and all login-authentications. Alarm provides the utility and functionality of muting and sending alarms and notifications. There will of course be some fail-safe methods to ensure that only non-vital alarms can be muted. Application interface handles the check on who is inside a work site, updates the speed of machines and updates location-data. Web interface, as mentioned earlier, is the part for the administrator where he can add, delete and edit user-information in the database.

• Database

- Component Interface Description

  The database, so far, is comprised of five tables; User, Worker, Copilot, ConstructionSite and AssocUandC (Associate user and copilot). User, being the base for all users, and Worker represents any user not associated with a Co-Pilot device (machine). All tables have their unique identifiers and Worker, Copilot and ConstructionSite have GPS-location-data in the forms of longitude and latitude.

  The relation between User and Worker will of course always be 1 to 1 and Worker is depending on User. For the table AssocUandC to have any data there needs to be data in Worker and Copilot and those relationships can be M to N.

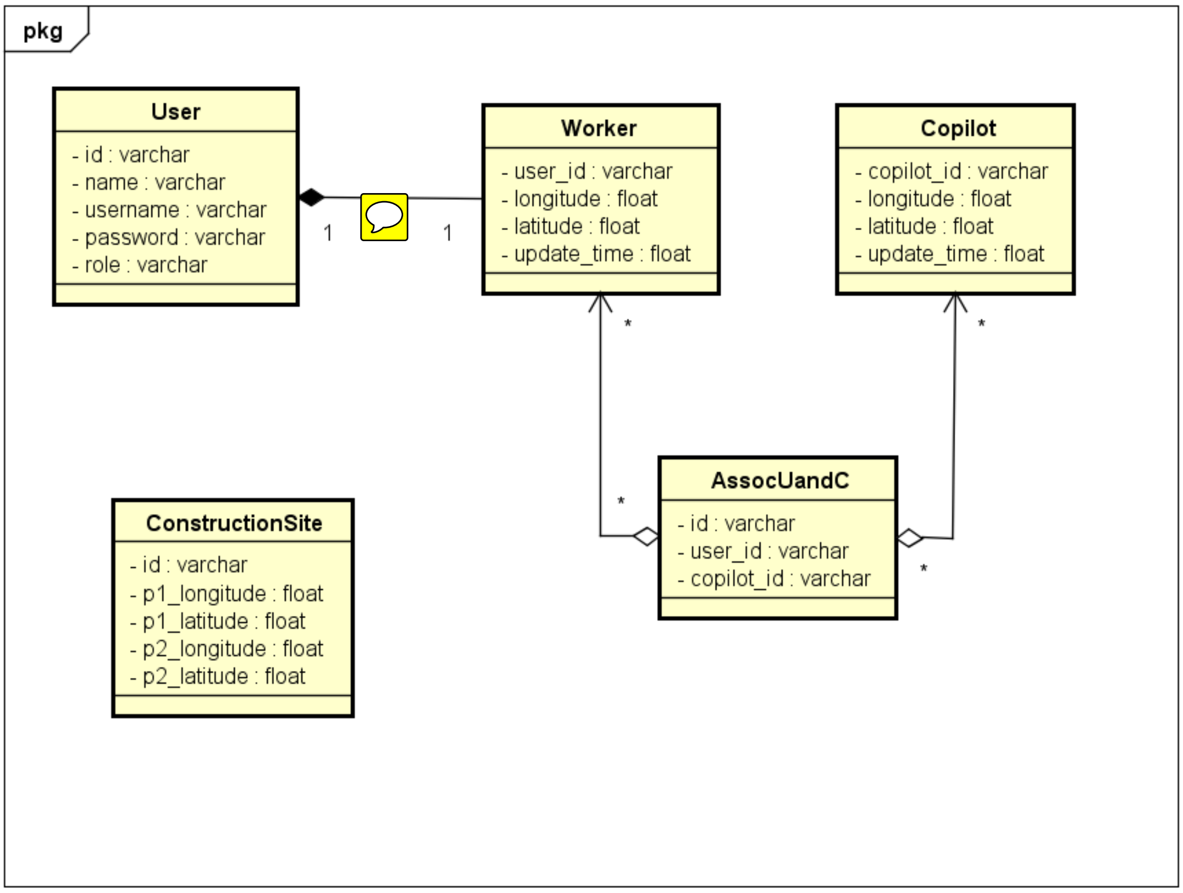


Figure 6: ER Diagram

- Component Design
  Initial database design is described in Figure 6.

4.2 Dynamic Behavior

Dynamic behavior of the system is displayed in Figure 7.

5 Graphical User Interface

5.1 Overview of the user interface

There will be an Android application for the user and web application for the administrator. The android application provides all of the users the possibility to login with their unique ID and password. By doing so, the application is able to get the users' exact location (longitude and latitude) and save them in the database, to be subsequently compared to other users' location.

The fist time the user logs in, a welcoming page is displayed and afterwards he can choose to exit the application - meaning it will run on background - or keep it open on the handheld device. The construction area location will be saved in the database. By being able to get the user location first, we check if the user is already inside a working area. In this occasion, we notify him with a pop up in the application with the message "You are now inside a working area.". Afterwards, we check and compare the worker and the machine operator distance. If they are too close, we notify them by a warning pop up alert with the message "Alert! Look out for vehicles.". In the case the worker exits the working area, a pop up message will be showed.

Regarding the administrator's interface, he will have a web application to manage the users. Basically, the web application provides the administrator with these functionalities: log in, log out, list all users,

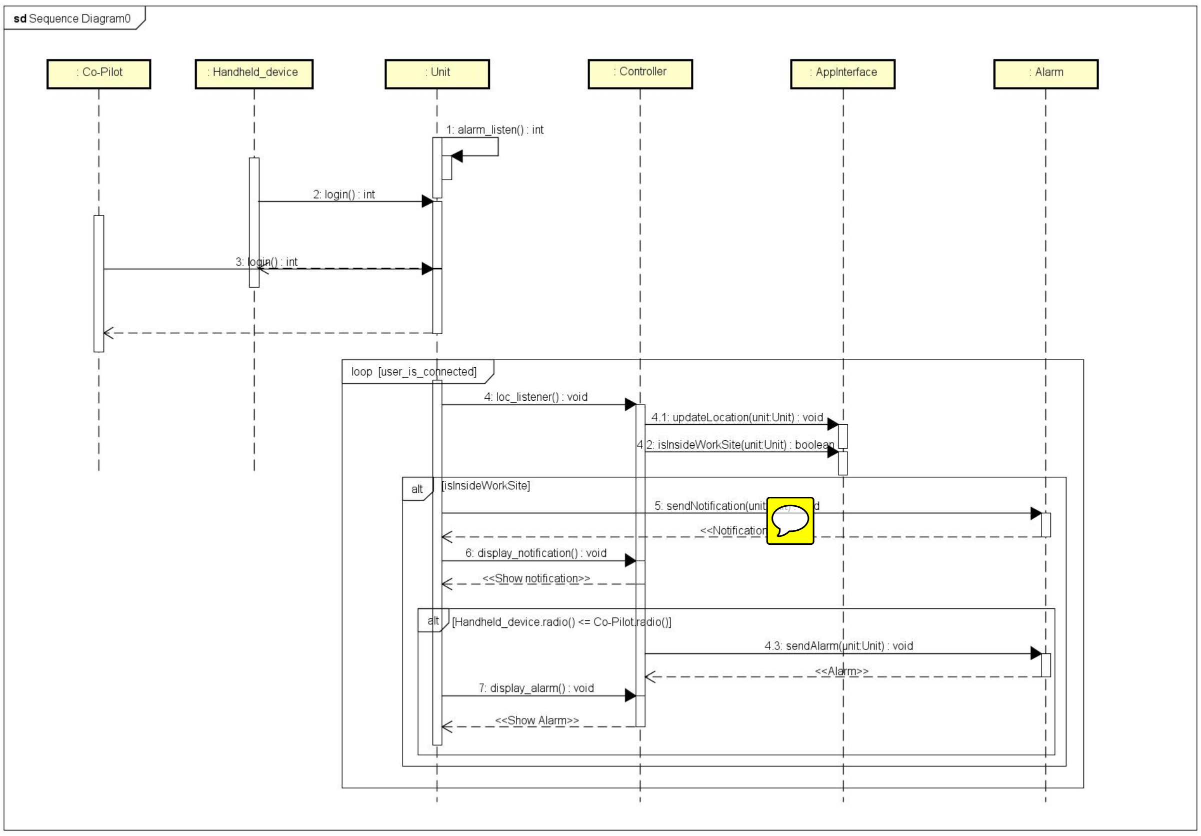


Figure 7: Sequence Diagram of Safe Assist

add users, edit user data and delete them. In an other phase, a map of the construction area and the workers on it can be displayed.

In the next subsection, from figure 13 to 15, the screenshots of how the web page of the administrator will look like are displayed.

5.2 Screen Images

In the following images, the first screen releases of how the android and web application will look like are displayed, together with a brief explanation, surely to be changed and updated in an other phase.

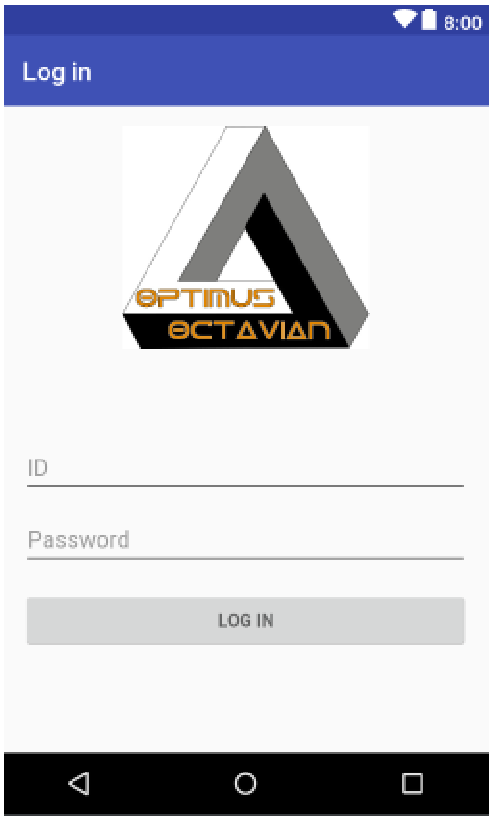
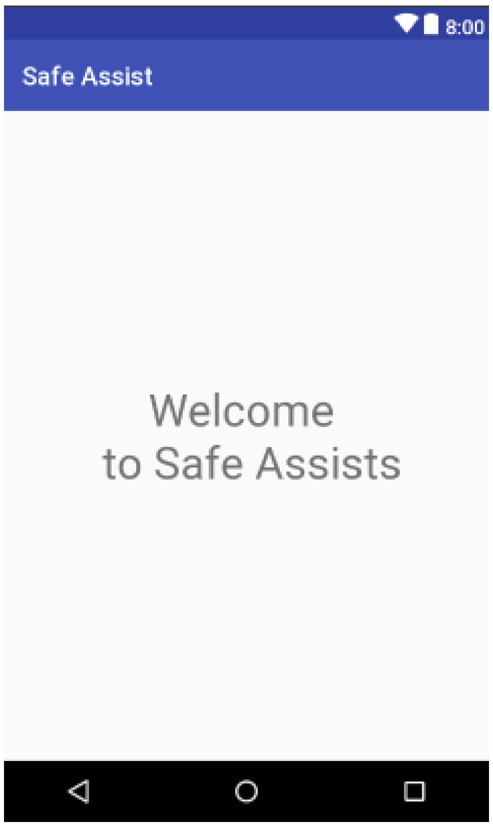


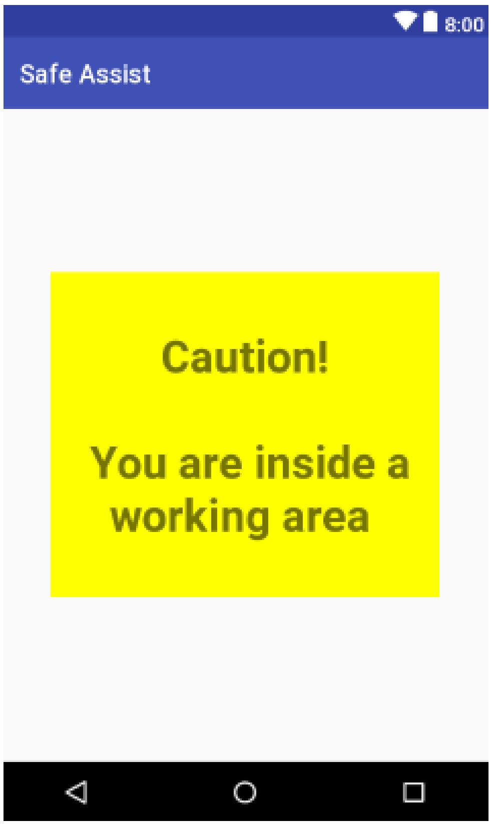
Figure 8: Login GUI

Figure 8 displays the log in view. By providing his/her unique Id and password the worker, machine operator or the manager will be able to login in the application so they can be observed to be safe during the working procedure.



Welcome
to Safe Assists

Figure 9: After login
Figure 9 displays the view after the user successfully logs in. The user is sent a welcoming page, and he can choose to exit the application and run it on background or keep it open on the hand held device.

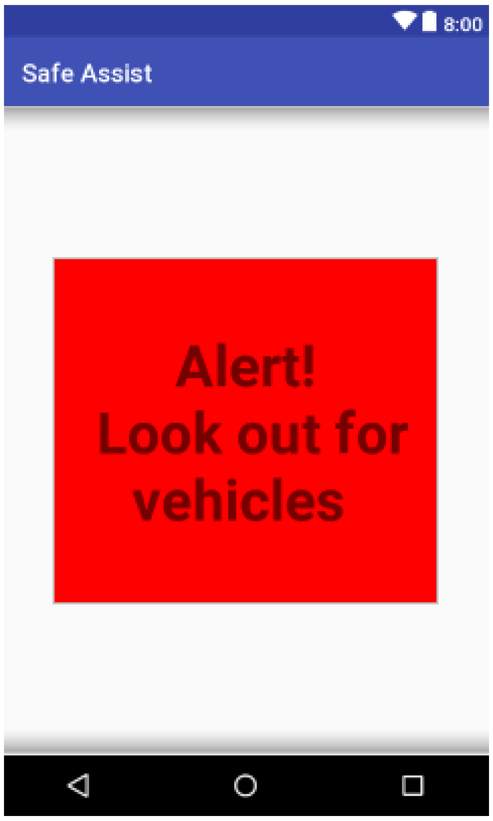


Caution!

You are inside a working area

Figure 10: GUI of entering the working area

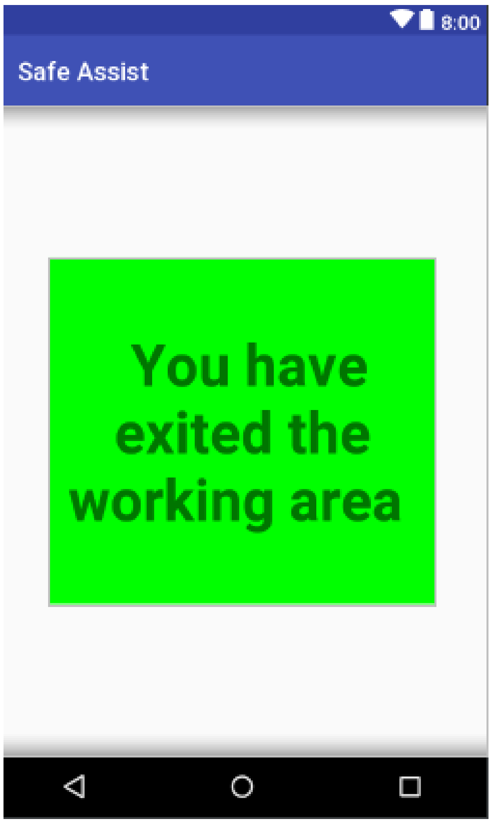
Figure 10 displays the application interface when the user is inside a working area. Since the user is logged in the application, we have the possibility to get his exact location in the construction area. Construction area will be set beforehand by the manager and its longitude and latitude will be saved in the database. We get the user (worker, machine operator or manager) location, and compare it to the construction area position. If the user is inside a construction are, we show him a pop up with the message "You are inside a working area".


Safe Assist

Alert! Look out for vehicles

Figure 11: GUI of warning the worker/operator

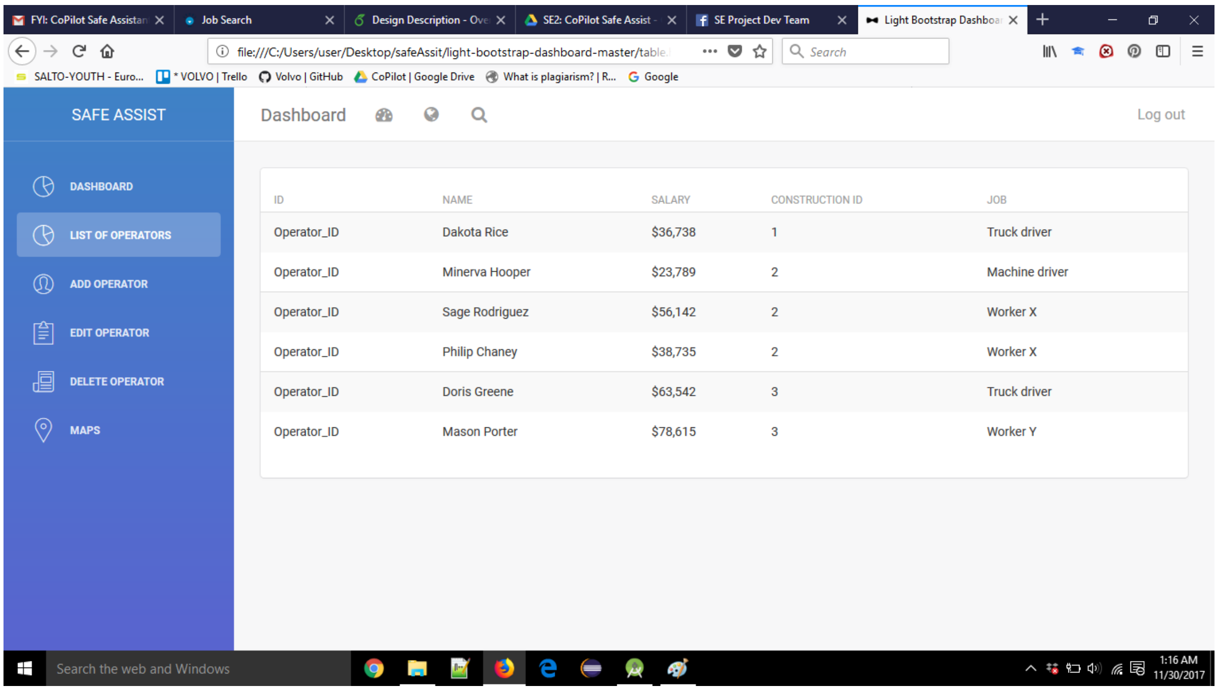
Figure 11 displays the application interface in case of close proximity to a machine. If the user is inside a construction are, we also compare his distance with the other machine operators on the working area. In the case that two users will be too close to each other and might be a risk, we show to both them a pop up with the message "Alert! Look out for vehicles".



You have exited the working area

Figure 12: GUI of exiting the working area

Figure 12 shows the display when exiting a working area. The pop up that will be showed in the case when a user exits the working area.



ID	NAME	SALARY	CONSTRUCTION ID	JOB
Operator_ID	Dakota Rice	$36,738	1	Truck driver
Operator_ID	Minerva Hooper	$23,789	2	Machine driver
Operator_ID	Sage Rodriguez	$56,142	2	Worker X
Operator_ID	Philip Chaney	$38,735	2	Worker X
Operator_ID	Doris Greene	$63,542	3	Truck driver
Operator_ID	Mason Porter	$78,615	3	Worker Y

Figure 13: List of all the workers - Administrator interface

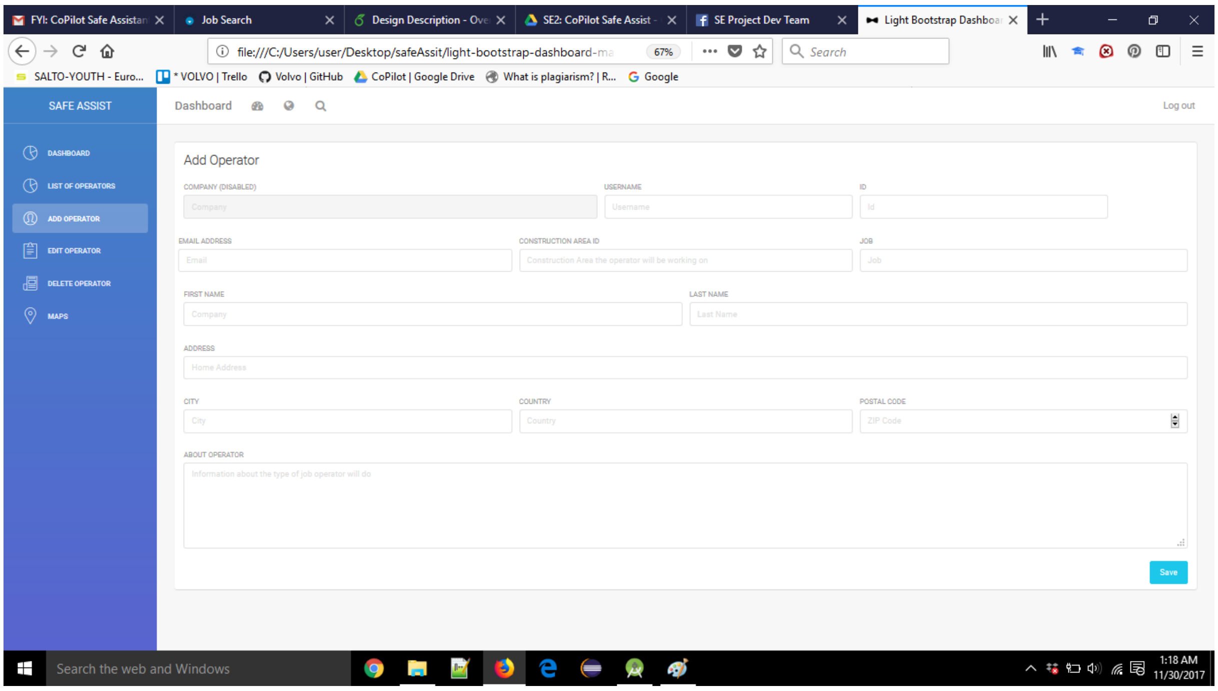


Figure 14: GUI of adding workers - Administrator interface

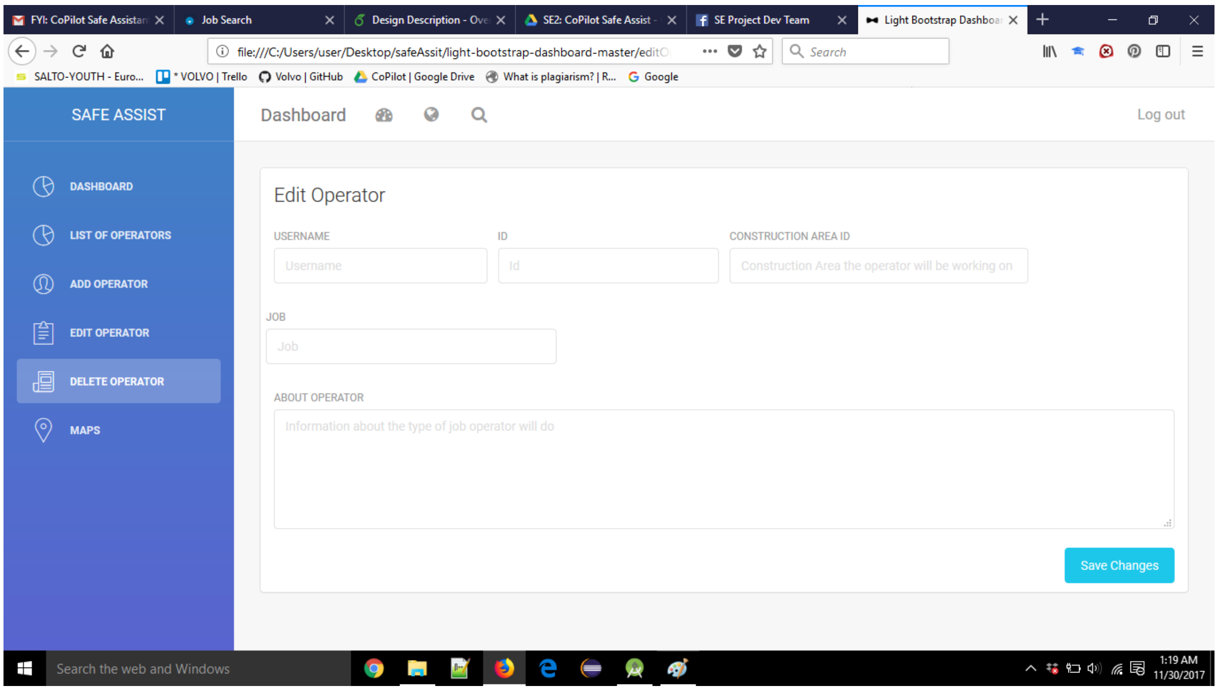


Figure 15: GUI of editing the workers data - Administrator interface

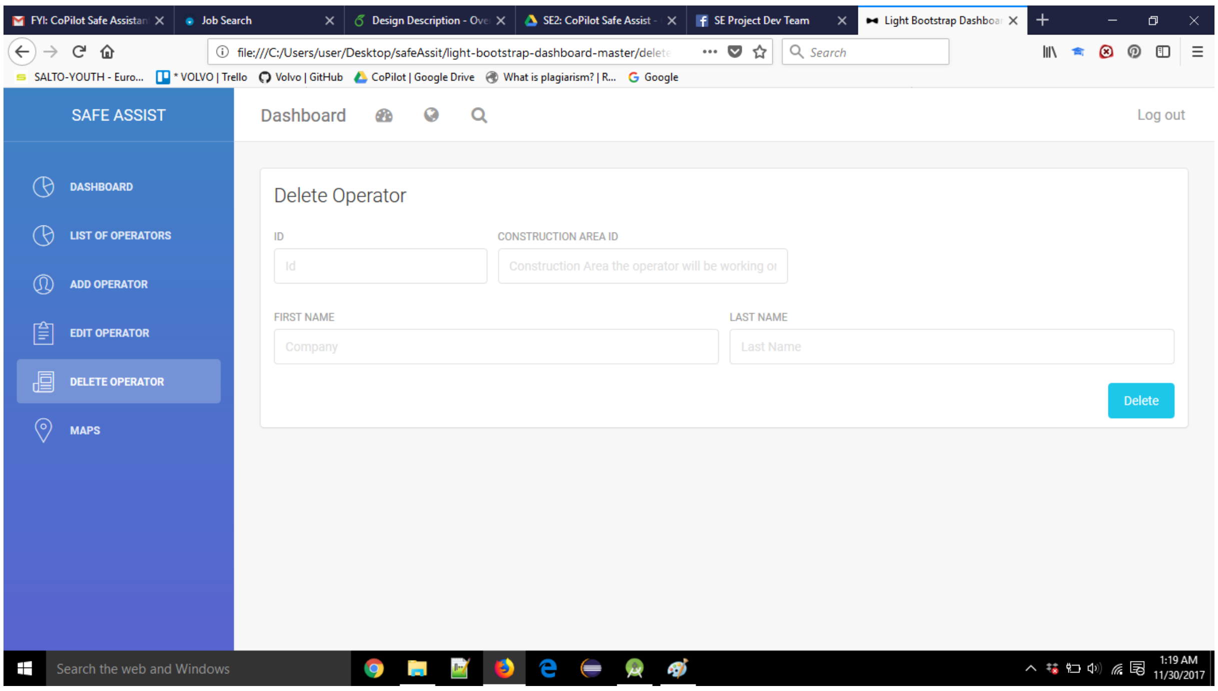


Figure 16: GUI of deleting workers from the working area - Administrator interface