

Project Proposal for an Evacuation and Emergency System

Contributors: Kacper Wołoszyn, Maciej Pikulicki, Pablo Valderrama Santiago, Denis Drobny, Alberto Perez, Thomas Cauneau, Kyrylo Shpak, Charlotte Cutler

Motivation

The main motivation for this project was, of course, to help ensure the safety of users. At this very moment, we have enough resources and technologies to implement this in real life. As of now, no such application appears to have been considered and there is a big market and range of consumers for this type of application. The idea of everyone having an easy way to escape from danger by just looking at their phone to show the quickest route, considering the popularity of phones now, is necessary.

Description

With this project, we want to help people to gain knowledge about escape routes and evacuation plans in different buildings and points around the city in case of emergencies. We plan to develop desktop application running on Java Virtual Machine that will be simulation (and very good base for later development) of Android mobile application

The main assumption is that our application gathers anonymous information about each user's position and movements. It will then analyze the data to work out optimal evacuation routes inside the buildings.

The system will analyze users' movements and positions during normal routine activities so as to detect the most crowded points and the users' ordinary routes inside the building; but it will also analyze this information during both emergency drills and real emergencies, obtaining very valuable data.

Users will be able to request this information in the app and the optimized routes generated will be provided to each user according to their position.

Certified registered building owners will be able to submit building plans so as to add new buildings to the system.

The system will have a data storage of all rooms and buildings and the statistics gathered during Test evacuations, to show and calculate the most convenient way of leaving the building at the time of an emergency.

Requirements

We are committed to getting the core of the desktop application on the right track. There is a possibility of complexity arising in this project, as there are many possible buildings, rooms and maps, so we need to be careful to keep the project as realistically feasible as possible in the time given to us. We are committed to documenting every aspect of the project and contacting the customer to see what they really want us to do, as well as maintaining talks with them to ensure that we are fulfilling the needs and wants of the person we are creating the application for.

In addition, if we have time, we are going to implement an environment for convenient use of the program (our commitment is to solve escape paths for the inputted map in a functional way that is not necessarily user-friendly).

We all agreeably decided to do this project in Java as everybody from our team know this programming language. After deeper conversation we gave-up the idea of doing this in Android because: 1st of all not all of us have those devices (some use iPhones OSX), 2nd of we don't feel to have enough experience to work on mobile environment. Nevertheless the main destiny of this application is to serve end-users on their smartphones with android operational systems. As it is commonly known that Android applications can be written in Java, we think that our project will be very good background and base to redevelop (transfer) it later from desktop to mobile devices (with proper funds, bigger team, and much more time) using our solution. Even though we are doing this as desktop project, we want to simulate mobile environment as much as we can, especially by creating similar graphical user interface and model.

During this project we are not going to connect application to real database. We will be mocking database connection by using text files with all necessary data. We are going to share those files via remote repository so that everybody is up to date with newest changes.

As mentioned above the graphical user interface of our Java program will simulate smartphone held in vertical orientation. Navigation over the application will remain mobile environment (so except developed inside-application buttons, there will be always available: back, home, and settings button just like in regular phone).

To find a shortest evacuation route we consider using Dijkstra's together with Dinitz or Ford-Fulkerson algorithm suited for our system. Also there will be graphical presentation of the computation (what end-user will see when asking for shortest evacuation way) .

We want to deeply describe the way to define completely different and custom situations so that the system will be flexible not only for pre-defined input data

What is more we decided to lead statistics about users who use our application. It might be very useful in further development and possible commercial use in the future. That is why we want to request everybody who uses our application to register and be logged in.

Also administration panel for administrator will be available to change mocked database from application level

Below we are listing our commitments and extension for the project:

Commitments

User can ask for a evacuation route.

User can visualize the route in form of a graph in the map.

Extensions

Owner of building can delete a plan.

Administrator can add or delete other users from application level.

Administrator can see the application statistics (e.g. average age of users, top 3 cities, top 3 buildings etc.)

Building owner can upload building plans with evacuation point from application level

Logged User can select building and current position in the building

Logged user can preview map of a building.

Logged user can log out.

Logged user can change password.

Registered user can log in.

Application visitor during registration process can fill in not-mandatory data for future statistics.

Application visitor can register onto the system as a normal user or building owner.

Logged user can compare the path that have been generated for him with the shortest possible, in the case that they are different, and see the bottlenecks on the shorter one.

Owner of building can modify a map.

Logged user can edit personal data.

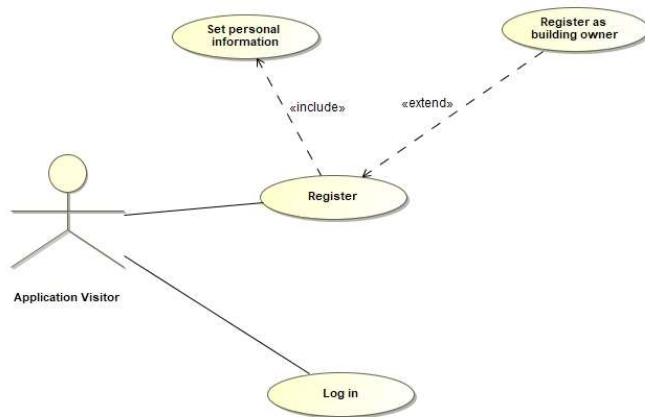
Administrator can modify users personal data.

System will display routes by using real building plan instead of graph

Use cases (below):

Application visitor (database associated)

package Data [AppVisitorUC]



Logged user (regular user, building owner, administrator) (database associated)

