POLITECNICO
MILANO 1863

Neighborhood Security

Software Design Document

Simone Ripamonti, Luca Stornaiuolo

June 20, 2017

Neighborhood Security Design Document
Version v1.0, June 2017

Contents

1 Introduction

1.1 About the Design Document

In this section we want to introduce briefly our application.

The application is developed for the course of "Design and implementation of Mobile Applications" at Politecnico di Milano. The goal of the course is to efficiently design and implement a mobile application on a platform of our choice. This documents illustrates the decisions we made in order to accomplish this goal.

This Software Design Document is a document that provides documentation that will be used as a overall guidance to the architecture of the software project. In this document we provide a documentation of the software design of the project, including use case models, class and sequence diagrams.

The purpose of this document is to provide a full description of the design of Neighborhood Security, a native Android application, providing insights into the structure and design of each component.

1.2 Platform

The choice of the platform was left to us. We decided to develop for Android.

Main reasons:

- Android native applications are written in Java, which is a language that is familiar to us
- We both use Android devices everyday, so we are familiar with the environment and we have the possibility to test the application on our physical devices

1.3 Choice of the application

We had full choice on the purpose of the application.

The choice fall on an application that could support associations like "Associazione Controllo del Vicinato", (volontari e specialisti volontari che forniscono consulenza e supporto alle Amministrazioni Comunali, alle associazioni locali e a privati cittadini che intendono sviluppare nel proprio territorio programmi di sicurezza partecipata e organizzare gruppi di Controllo del Vicinato). The application gives the possibility to alert other user about criminal events that have just happened.

The particulary that characterises this application is the possibility to receive notification about events in the user's favourite locations in real time and to brows a map of all the signaled events.

The idea for the application came from personal experience. In our towns, groups of people are currently reporting criminal events using Whatsapp group chat, but this is not user friendly:

- You need to know and contact one of the administrators of the group
- You need to share your personal phone number and information
- You cannot have an ensemble view of all the reported events

1.4 Risk analysis

During the problem analysis, we identified some risks that can compromise the correct development of the project.

We had to be very careful during the requirements collection phase, because the requirements must be clear so to avoid delays due to misunderstandings.

Modelling formally the application allows the reader not to be confused by it and learn requirements in a clear way.

Another possible delay was learning Android concepts or techniques in advanced stages of the project, leading to possible code rewriting and inevitable loss of time.

1.5 Time constraints

Our time constraints were not as strict as they would have been in a project developed for real stakeholders. We had no precise and punctuated deadlines, but to deliver the project among the different call dates for the course.

We begin to develop our application at the beginning of the second semester). Developing, testing and creating documentation took 3 months. Even the server side part was built and manteined during the same time span. We started development at April 2017 and complete it in July 2017.

The team is composed by two people with both solid Java knowledge but little experience in Android, thus we had to dedicate some time to learn the environment.

1.6 Stakeholders

The main stakeholders of our project is the professor and the other students attending the presentation. The audience wants to have a clear idea about the project idea and realization. Professor's main goal is to check if the concepts taught during the course are clear to us and if we had success in implementing them in our project.

Even though we had not currently planned to release our application in the Google Play Store, we designed it keeping simple and intuitive, suitable for every kind of users.

We use English as main language for our project, due to its diffusion in the world, and we currently provide an Italian translation. If we ever decide to officially release *Neighborhood Security*, we will introduce other major languages to make it more usable.

2 General overview

2.1 Idea

Neighborhood Security is a native Android application that provides users the possibility to report criminal events and receive notification about favourite areas.

The application does not require users to be authenticated in order to browse the reported events, indeed everyone can easily browse the reported events over a map and easily understand the safety of a certain location by simply checking how many reports have been done and of which kind.

Users can register an account, using a combination of email and password or using their Facebook or Google account. Registered users have access to a new set of functionalities, such as subscribing to notification about a certain location, reporting events and voting already submitted events. Once subscribed to a location, users can now receive real time notifications as soon as another user report a criminal event.

The main components of the project are:

- A server side part, composed by a relational database that contains events and subscriptions and the application logic that is in charge of notifying interested users about events;
- A client side part, the Android application, which tasks are to query the relational database, show the gathered informations and provide users a point of interaction with the service.

2.2 Core features

Here we list the core functionalities that can be found in our application. We decided to divide them considering the fragment in which is organized, allowing the reader to easily understand them and where they could be found in the application.

- Home
  - First point of interaction with the user. It simply shows the user the two main relevant features of the application: seeing reported events and subscribe to notifications.
  - A lateral menu is provided to allow the user to access to other features, first of all the creation of an account that gives access to the interesting features of the application.
  - A set of icons provides the user with insights about his usage of the application.
- Authentication
  - Allows the user to authenticate using three different ways: a combination of email and password, a Google account or a Facebook account.
  - If they choose email and password authentication, users also have the capability to reset the chosen password in case they have forgotten it.

  - Choosing the other methods, users are required to allow the application to access to their personal information such as profile information and email address

- Event map
  - Provides users the possibility to see reported events placed on a map. Different kind of events are represented by means of different marker icons.
  - If the user gave permission to access location informations, the map is centered in the current position.
  - By clicking on a marker, user can see more detailed information about the reported event
  - In order to decrease the number of markers placed in the map, events are clustered according to their position. By clicking on a cluster, users can see the list of events that are contained in that cluster
  - A search bar is provided in order to easily search and move to a particular location, with an auto-complete feature provided by Google

- Event list
  - Shows a list of events that can be refreshed if the list is related to a specific location, subscription or submitting user.
  - User can easily understand the kind, by looking at characteristic icon, the location and date of the event.
  - By clicking on the event, a new view containing its details is shown
  - If the user is the creator of a particular event, he can delete it by simply long clicking on the event

- Event creation
  - Only authenticated users have access to this feature
  - Events are categorized according to the kind of event: burglary, ... etc
  - Users must insert a brief description of the event
  - Position of the event can be chosen by inserting the place name (using Google Places auto-complete service) or by inserting pure coordinates, that can be obtained by using the location service if authorized.

- Event detail
  - Shows all the informations available about the selected event. Reporter name is never shown to other users by design.
  - A small map allows users to understand more precisely where the event happened.
  - Authenticated users can vote the event. The more an event has been voted, the more we consider reliable the reported event.

- Subscription list
  - Shows a list of subscriptions, if any, of the currently logged in user. The user can enable or disable notifications about a particular subscription and delete them if he is no longer interested in.
  - By clicking on a subscription, a new view containing the list of events that match the subscription is shown

- Subscription creation
  - Only authenticated users have access to this feature
  - Subscription are characterized by two components: a position, that can be a place or GPS coordinates, and a radius, up to 2000 metres.

2.3 General qualities

2.4 Functional requirements

2.5 Non-functional requirements

3 Data design

3.1 Internal software data structure

3.2 Database design & implementation

4 Architectures and component level design

4.1 System architecture

4.2 Architectural design

4.3 Java package organization

4.4 Security

5 User interfaces

5.1 Splash screen

5.2 Home and drawer

5.3 Authentication

5.4 Map

5.5 Subscription list

5.6 Subscription creation

5.7 Event list

5.8 Event creation

5.9 Event detail

6 External services

6.1 Google Firebase

6.1.1 Authentication

usato per consentire l'autenticazione tramite facebook, google e email

6.1.2 Cloud Messaging

usato per consentire l'invio delle notifiche ai disposiivi da parte del web service

6.1.3 Job Dispatcher

usato per schedulare ogni mezzanotte la rimozione di eventi e sottoscrizioni che non sono state refreshate da più di 7 giorni, per rimouvere eventi/sottoscrizioni eventualmente rimosse

6.2 Google Play Services

6.2.1 Authentication

usato per l'autenticazione tramite firebase

6.2.2 Location Places

usato place autocomplete per semplificare la segnalazione di event, creazione di nuove sottoscrizioni, ricerca eventi, posizionamento mappa

6.2.3 Maps

usato per mostrare mappa degli eventi e come header nel dettaglio evento

6.3 Facebook

usato per l'autenticazione tramite firebase

6.4 Neighborhood Security Webservice

rest api per gli eventi

6.5 Other minor services

glide, altre librerie

7 UML diagrams

7.1 Use cases diagrams

7.2 Class diagrams

7.3 Sequence diagrams

8 Test cases

9 Cost estimation