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

2.2 Core features

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