

Proceeding

A Smartphone Application for Supporting the Data Collection and Analysis of the Cultural Heritage Damaged During Natural Disasters [†]

Antonio Costanzo^{1,*}, Sergio Falcone¹, Rosalbino Bisignano², Michele Straface², Giuseppe Ritacco², Carmelo La Piana¹, Massimo Musacchio¹, Francesco Calimeri²

¹ Istituto Nazionale di Geofisica e Vulcanologia, Italy; sergio.falcone@ingv.it (SF), carmelo.lapiana@ingv.it (CLP), massimo.musacchio@ingv.it (MM)

² Department of Mathematics and Computer Science, University of Calabria, Italy; bisignano93@yahoo.it (RB), mstraface94@gmail.com (MS), giuseppe.ritacco93@gmail.com (GR), francesco.calimeri@unical.it (FC)

* Correspondence: antonio.costanzo@ingv.it (AC). Tel.: +39-0984-496068

† Presented at the 4th International Electronic Conference on Sensors and Applications (ECSA 2017), 15–30 November 2017; Available online: <https://sciforum.net/conference/ecsa-4>.

Published: 14 November 2017

Abstract: The adverse impacts of natural disasters on lives and livelihoods, as well as regional and local economies, are increasingly evident, and losses to both tangible and intangible cultural heritage due to these disasters pay an important role in the total amount. In fact, damages to sites, structures and artifacts of cultural and historical value, as well as impacts to cultural tourism and the financial resources, produce a strong competitive disadvantage on local communities. Emergency decision making, based on awareness of the suffered damages, can play a crucial role in the attempts of improving resilience of the strategic elements; however, this process typically requires a fast overview on large territories. In this work, we propose a novel framework for obtaining an agile solution to quickly collect and analyze picture galleries and information provided by both internal staff and citizens through commercially available mobile devices. This solution virtually generates a network of information sources during emergency time (e.g., a seismic sequence), and allows to produce a situation map in GIS environment, hence supporting the health status analysis of cultural heritage over time. This paper presents the prototype system composed of: (1) a smartphone application for the acquisition of new information and the examination of existing one; (2) a web-service for exchanging data with databases, and (3) a local service that makes use of a proper piece of software for obtaining a 3D reconstruction from new picture galleries. The proposed system results in a scalable, exportable and modular tool useful during the emergency and for preserving memories of local communities.

Keywords: Cultural Heritage; Natural disasters; Smartphone application; 3D reconstruction; Network of information sources; Support for emergency decision making

1. Introduction

The losses due to natural disasters in terms of lives, livelihoods and built environment, often severely impact on the life of local communities. These negative effects severely strike not only local and regional economies, but also culture and memory. In such contexts, tools for supporting decision-making process are fundamental, both during and after the emergency phase. In the last decades, development of powerful mobile devices and geospatial technologies allowed the use of information application in the disaster management field [1,2].

In this work we present a prototype of a post-disaster information system, suitable to generate a network of sources during emergency time (e.g. a seismic sequence), and to support the health

status assessment of built environment over time, with particular regard to cultural heritage. The main strength points consist of the possibility to directly and simply share input data gathered both from experienced staff and citizens, and outputs via mobile App and Web portal.

2. Methods

2.1. Requirement Analysis

Before the design and development phase, requirements related to the information management and system functionalities were analyzed for stakeholders, as well as documentation of similar solutions [e.g. 1,2]. Main goals and key functionalities are reported in Table 1. In general, stakeholders demand computer systems able to collect, share and elaborate informative layers (pictures and metadata, essentially) about damages to buildings and facilities along with environmental effects induced by natural disasters (e.g. earthquakes) or human actions. In other words, the main functionalities required are: (1) images acquisition and three-dimensional reconstruction of buildings (or parts of them) in order to support the "health status" assessment (e.g., it is useful for Cultural Heritage after a strong earthquake); (2) the acquisition of information about the induced-disaster effects to map the damage levels. Data can be sent according to user type/permissions (i.e., experienced operator or citizen), also to assess the information reliability. Nevertheless, access to collected data was granted to all users in order to foster the sharing among wider communities. A large number of interactions and information communications between mobile/web clients and back-end services were expected for exchanging pictures, textual data, 3D object and metadata.

Table 1. Key functionalities identified through the requirement analysis.

Component	Functionally	Goal
Back-end services	Automatic 3D Reconstruction	Generate a three-dimensional model from a photographic dataset
	Storage and cataloguing	Keep the information acquired and make it easy to consult
	Authorization procedure	Enhance the reliability of the obtained data
	Generate reports of the macro-seismic surveys ¹	Complete sheets of damaging for the cultural heritage
Mobile and web client	Report disaster-induced damages	Save information and pictures related to disaster-induced damage
	Search nearby buildings	Retrieve buildings near the user
	Visualization of the geodata	Display the base map and overlaid damages.
	Geolocation service	Retrieve user location information, such as latitude and longitude
	Access to the camera (mobile client only)	Take pictures using device cameras or retrieve pictures from the image gallery.
	Working in offline mode (mobile client only) ¹	Allow the usability also without internet connection

¹ Functionalities not still implemented in the presented version of the information system

2.2. Architecture Overview

Accessibility and operability are essential requirements for a disaster management system under extreme situations, in which reliability and speed of network connectivity are typically different than in “normal” situations. In addition, a system able to work across multiple platforms with affordable cost is a great advantage, in order to expand its user community. Based on these considerations, the generic system architecture (Figure 1) is outlined with 3 blocks:

- The first level of the system consists of clients for data capturing and for displaying processed results;
- The second level represents a web service able to store the data on file systems and/or spatial databases;
- The third level is a local service to manage the automated 3D reconstruction by photo gallery.

Details of these components will be illustrated next.

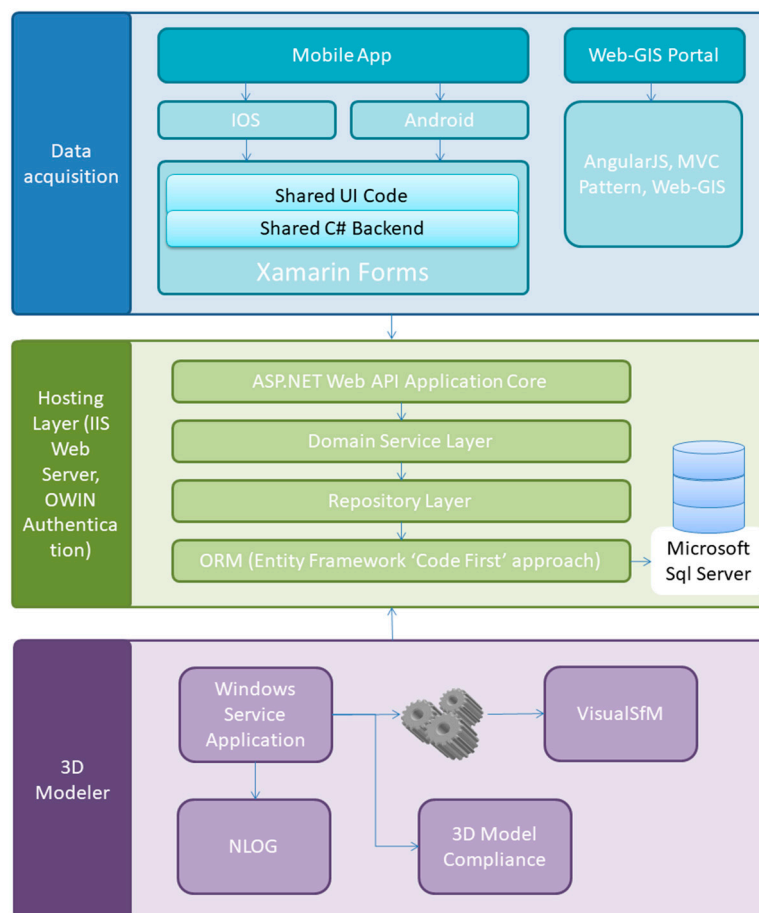


Figure 1. System architecture: data acquisition (upper panel), hosting layer (middle panel) and 3D modeler (lower panel).

3. Results and Discussion

3.1 Client Application

In this paragraph, we illustrate the mobile application; the Web-GIS portal is still under development.

The chosen mobile App development environment is Xamarin [3], a framework for native and/or cross-platform App in C#. The framework is based on open source Mono (a framework for

implementing .NET Framework of Windows on Linux platform), that offers full support to Android e iOS platforms.

Essentially, Xamarin is composed by:

- Xamarin.Android for developing Android Apps through wrapping native API in C#;
- Xamarin.iOS for developing iOS Apps through wrapping native API in C#;
- Xamarin.Forms, for sharing components and code.

Thanks to these components, Xamarin allows to manage all Android and iOS features, from UI to the hardware resources.

Figure 2 shows the main functionalities of the proposed App. When the mobile client is launched on a mobile device, the homepage with two buttons is shown in order to choice: (1) the visualization of points of interest (POI) or (2) the compilation of the questionnaire (Fig. 2a).

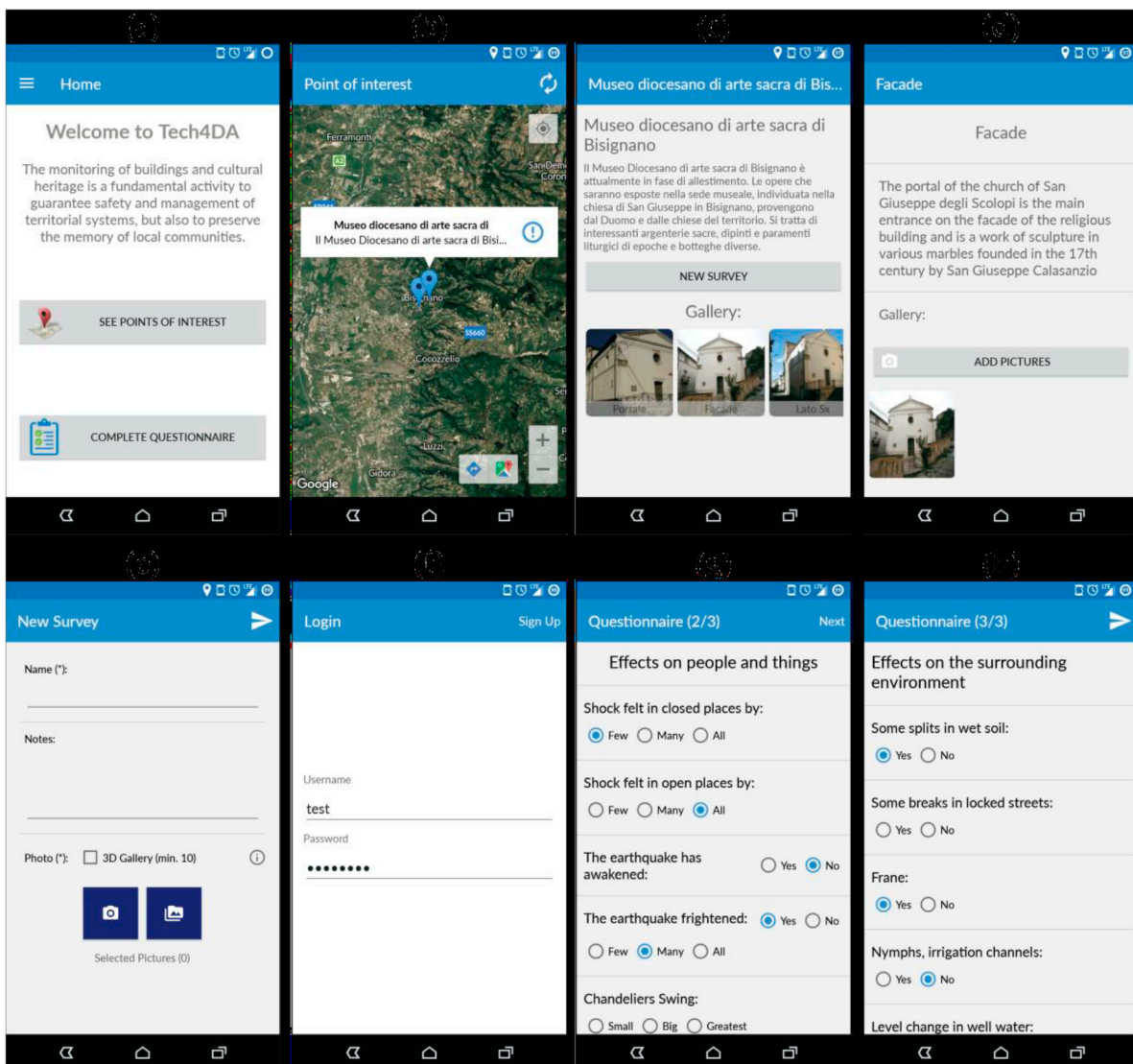


Figure 2. Mobile application: homepage (a), points of interest (POI) on map viewer (b); page of POI details (c); details about an existent photo gallery (d); page to acquire a new survey (e); login form (f); questionnaire pages about the effect due to an earthquake on people (g) and environment (h).

By clicking on "See Points of Interest", the censured POI are reported on the map viewer; in particular, the mobile client makes a request to the server for searching POI in a circular area around to the user (Fig. 2b), based on centre (latitude and longitude are provided by GPS of the device) and radius (fixed at the moment, but it should be user-defined in next releases). By touching a pin marker

on the map (see Fig. 2b), the name of POI and first 40 characters of its description are shown. Once the user clicks on the information icon, the page of POI is loaded. This page contains the whole textual description and the photo galleries; moreover, a button to acquire new images is available (Fig. 2c). The user can add pictures to an existing galleries, directly by selecting it, or a new photo gallery by clicking on the button. The first choice leads to the page in Figure 2d; instead, the acquisition of a new photo gallery can be made through the activity shown in Figure 2e, where the survey name and any notes must be introduced before taking or uploading pictures. If the user provides at least 10 pictures, he must also indicate whether these are useful for 3D reconstruction (cf. Fig. 2f). However, prior to the acquisition phase, the login page is displayed to the user, who is prompted to enter their username and password. If the user information are correct, the acquisition phase is allowed.

On the main activity, if the user clicks on "Complete questionnaire", upon login proper pages are loaded, reporting questions about the effects of an earthquake both on people (see Fig. 2g) and environment (see Fig. 2h).

3.2 Hosting Layer

3.2.1 Storage

The storage space to collect information and images is hosted on a server machine with dedicated hardware architecture. The server machine feature 4 cores clocked at 2.50 GHz and 8.00 GB of RAM, and runs Windows Server 2016 Standard Edition, 64bit version. The virtual machine can be upgraded at any time. The vector data are stored via a Microsoft SQL server [4] database and managed through a specific web service, also, a GIS spatial engine provides support for the use and management of geographic objects.

Storage is organized in a directory tree: the root directory is "repository", the second level is "POI" and the third level is "gallery"; inside this latter directory there are pictures sent by mobile app. "id_picture+name_picture.jpg" and "../repository/POI/id_picture+name_picture.jpg" are examples of name and URL for downloading of an image, respectively.

3.2.1 Web Service

The web service plays the key role of the communication channel between the database and the data collection devices. This service was developed through the Web Api template of Microsoft Visual Studio IDE [5] and Microsoft .NET Framework [6]; whereas, the management of the database was guaranteed by Entity Framework [7] and Microsoft SQL Server [4].

The Code First [8] of Entity Framework is used to create the database tables. This approach is mainly useful in Domain Driven Design. In fact, with the Code-First, we focused the efforts on the domain design and start creating classes according to domain requirement, rather than designing the database first and then creating the classes. The database was created according to the entity classes and their configuration. The main task of a web Service is to handle HTTP(s) requests; this means performing CRUD (Create, Read, Update, Delete) operation on the database. Each of the class, referring to a database table, must be associated to a "Controller" class [9].

A controller class contains the implementation of the HTTP methods (Get, Post, Put, Delete) to interact with the database. All controller classes share the CRUD standard operations.

Particular cases concern:

- for classes returning a POI list, a method to get objects into a circular area defined by centre and radius;
- for classes saving images on storage, a method to convert and save images both with original and thumbnail size;
- for classes returning the municipality, a method to detect the administrative territory by latitude and longitude of user.

To make the output of the different controller classes uniform, a common object was defined in order to represent data. The "Asp.NET Identity" authentication system, based on "OWIN" framework and "OAuth" protocol, was implemented for managing the data access [10].

3.3. Local Service for 3D Reconstruction

A local service was implemented for automating the visual 3D reconstruction process based on image elaboration. In particular, this Windows service is installed on the server machine as interaction tool between 3D reconstruction software and database. For this purpose, also, the open source software VisualSfM [11] is installed on the server. VisualSfM is an application for 3D reconstruction that makes use of structures from motion technique [e.g. 12]. In order to boost performance, this software exploits multicore parallelism for feature detection, feature matching, and bundle adjustment[13].

Hourly, the local service check if new photo galleries are available, in positive case:

1. VisualSfM is called and pictures are retrieved;
2. once homologous points are detected on the pictures, the 3D modeling is performed;
3. at the end of the process, a control is executed to check that only one model was produced;
4. a variable is initialized about the elaboration state of the photo gallery in the measures table.

Moreover, a daily log file referred to the executed operations was implemented through Nlog [14] registration platform.

An example of input data and outcome obtained by 3D reconstruction process, managed through the local service, is plotted in Figure 3.

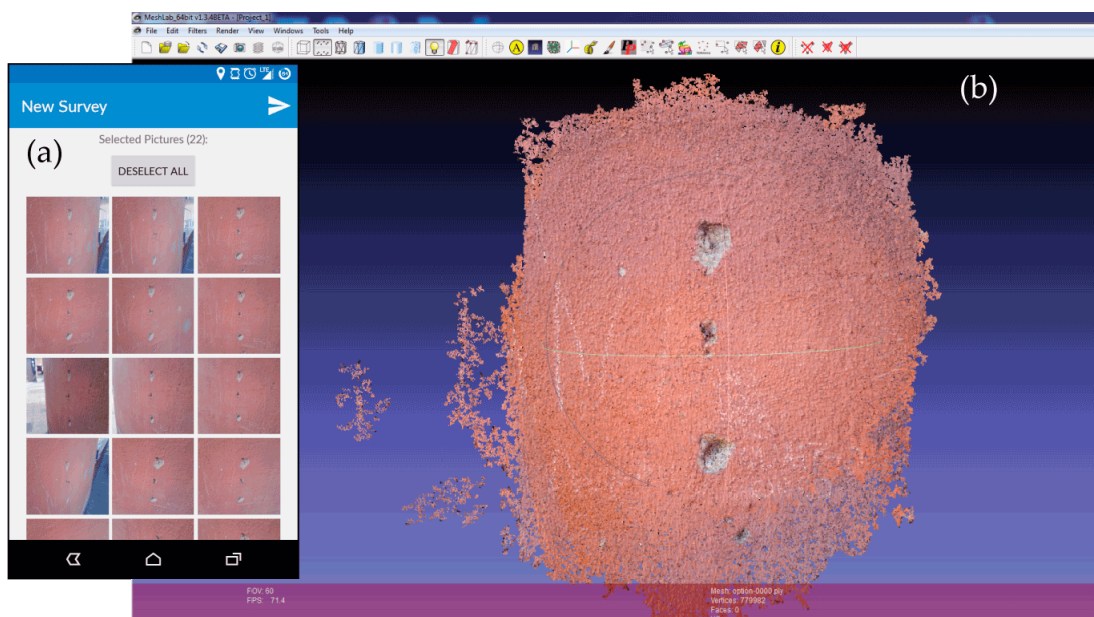


Figure 3. Windows service for managing the 3D reconstruction process: input pictures of a plaster detachment acquired by mobile App (a) and model visualized through Meshlab software [15] (b).

4. Conclusions

An information system for supporting the damage assessment during (and after) disaster-induced emergency phases was designed and implemented. We presented here the main components of such system: (1) mobile application to gather/share data from/via experienced staff and citizens; (2) the web service able to manage exchanges between the devices and database; (3) the Windows Service to control the 3D reconstruction process based on collected pictures.

The presented system version is in prototypical form; hence, before making the Application public further tests are needed. Furthermore, in the next release some improvements will be

introduced: a Web-GIS portal as new client tool; the offline mode for mobile App; an integrated 3D model viewer both for mobile App and Web portal; the reports of damaging related to the buildings and/or cultural heritage.

Acknowledgments: The authors thank INGV QUEST Team for the questionnaire about the effects due to earthquake and suggestions for implementing the related pages of the mobile App.

Author Contributions: In this article: AC and SF wrote the manuscript, CLP, GR, MS, MM, FC attended in drafting; AC, SF and MM conceived architecture of the information system; mobile application, web service and local service to manage 3D reconstruction process were implemented by RB, MS and GR, respectively, SF, AC, FC supervised these work packages; All authors give final approval of the version submitted.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. He, Y., Zhang, D., Fang, Y. Development of a mobile post-disaster management system using free and open source technologies, *Int J Disast Risk Re*, 2017, 25, 101-110, ISSN 2212-4209, <https://doi.org/10.1016/j.ijdr.2017.08.007>.
2. Aydin, C., Tarhan, C., Ozgur, A.S., Tecim, V. Improving Disaster Resilience Using Mobile Based Disaster Management System, *Proc Tech*, 2016, 22, 382-390, ISSN 2212-0173, <https://doi.org/10.1016/j.protcy.2016.01.027>.
3. Mobile App Development & App Creation Software. Available online: <https://www.xamarin.com/> (accessed on 20 October 2017)
4. Microsoft SQL Server 2016. Available online: <https://www.microsoft.com/it-it/sql-server/sql-server-2016> (accessed on 20 October 2017)
5. Microsoft Visual Studio IDE. Available online: <https://www.visualstudio.com/> (accessed on 20 October 2017)
6. Microsoft .NET Framework. Available online: <https://www.microsoft.com/net/download/framework> (accessed on 20 October 2017)
7. Entity Framework. Available online: <https://docs.microsoft.com/en-us/ef> (accessed on 20 October 2017)
8. The Code First of Entity Framework. Available online: <http://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx> (accessed on 20 October 2017)
9. Freeman, A., Your First MVC Application. In *Pro ASP.NET MVC 5*, 5th ed., Apress®: New York, USA, 2013, pp. 11-49, ISBN 978-1-4302-6529-0, 10.1007/978-1-4302-6530-6.
10. Asp.NET Identity. Available online: <https://docs.microsoft.com/en-us/aspnet/identity/overview/getting-started/introduction-to-aspnet-identity> (accessed on 20 October 2017)
11. VisualSfM. Available online: <http://ccwu.me/vsfm/> (accessed on 20 October 2017)
12. Szeliski, R. Structure from motion. In *Computer Vision*, Springer, London, UK, 2010, Volume 1, pp. 303-334, ISBN 978-1-84882-934-3, https://doi.org/10.1007/978-1-84882-935-0_7.
13. Wu, C., Agarwal, S., Curless, B., Seitz, S.M. Multicore Bundle Adjustment. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, 20-25 June 2011, Colorado Springs, CO, USA; IEEE, New York, US, 2011, 141, pp. 3057-3064, 10.1109/CVPR.2011.5995552.
14. Nlog. Available online: <http://nlog-project.org/> (accessed on 20 October 2017)
15. Meshlab. Available online: <http://www.meshlab.net/> (accessed on 20 October 2017)



©2017 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).