MobileSage Deliverable D2.4: System Requirements Specification for Content Management Service

Authors:
Till Halbach Røssvoll
Valentin Alonso Gracia

Version: 1.0
Date: 2012-04-10
Dissemination Level: Public

Abstract

MobileSage's Content Management Service helps making content available and distributes it for all people in the need for it, and with an enabled smartphone. The service is therefore an enabler for the self-serve society by allowing to produce a mixture of professional and user-generated content, nicely following the help-for-self-help principle.

This deliverable documents the system and lists all mandatory requirements for its implementation. The requirements are derived from the users requirements which in turn have been determined in focus group discussions.

Table of Contents

1 Introduction

This section details the background and scope of this Deliverable.

1.1 Background

This document defines the Content Management (CM) functional specifications for the scenarios planned for within the MobileSage project. With these requirements at hand the basics for the implementation of the CM Service are available.

MobileSage is an Ambient Assisted Living project (AAL) [1]. The main objective in MobileSage is the writing of two service applications [2]. The first is a mobile client application, called *Help on Demand Service*, and the second is a Web server application, here referred to as *Content Management Service*.

1.2 Scope of the Deliverable

This Deliverable specifies the system requirements for the MobileSage software deliverable Help on Demand Service and the interface between Help on Demand Service and Content Management Service.

The system requirements are based on *MobileSage Deliverable D2.2: User Requirements Specification [3].* They serve as input for subsequent project phases, including *Prototyping* and *System Implementation* and *Integration.*

The system requirements of this Deliverable are complemented by the system requirements in *Deliverable D2.3*, which are valid for MobileSage software deliverable Help-on-Demand Service [4]. It is stressed that the requirements are not mandatory for the implementation. Rather, it is recommended to deploy a continuous triaging process of the requirements by means of a suitable requirements or issue tracking tool.

All sections shall be regarded as formative sections except the Mockups section, which is informative only.

2 System architecture

The overall system architecture is shown in Illustration 1 in the System Requirements Specifications for Help on Demand Service [4]. The main components of Content Management are:

- Dialog Manager
- Content service
- Storage service

A user with multimodal *Content* may use a *User Agent* like a browser to gain access to the Content Management Service, provided as a *Cloud Service* on the *Network* (the Web). The user

interacts with the *Dialog Manager*, which in turn controls the *User Interface*. The main logic for handling the multimodal content lies in the *Content Service*, which has a modular design to be able to add content types, such as Haptics, Geographical data, etc., in a simple way. MobileSage includes the following modules:

- *Video*
- *Audio*/Speech
- *Subtitles*
- *Text*

The Content Service is responsible for aggregating content from different sources, including local services like Storage Service, and remote ones like geographical services such as Google Maps, traffic timetables, and others. The task of *Storage Service is t*o store fabricated content locally.

The Content Service also handles machine requests for content. To ask for content from the CM Service, the Help-on-Demand Service prompts the Content Service, which in turn sends out search requests to registered content type services, handles the responses, and collects and aggregates any help content. The content is then directly conveyed back to HoD.

3 Choice of technology

The system as sketched above is basically a content management service (CMS). There exists a number of readily available CMSs with various functionality, such as Wordpress, Drupal, Joomla, eZ, Plumi, just to name a few. These existing solutions do not provide the desired functionality for MobileSage; however, it is viewed as being a bigger effort to build a CMS from scratch than extending an existing CMS where a great deal of the basic functionality like file uploading and user management already is implemented. Moreover, it is crucial that the system can be extended into various directions in a later production system.

The system will hence be implemented by means of the following building blocks:

- Drupal CMS (based on the PHP scripting language)
- Apache web server
- MySQL/Postgre database

The technoloy choice is based on the following reasoning:

1. Support of all major CM features
2. High maturity
3. Robustness

4. High popularity of and good support for all major building blocks
5. Rich variety of available additional PHP libraries and modules
6. Good scalability
7. Rich functionality of additional CMS modules

4 System requirements

The system requirements are split into three logical parts regarding UI, functionality, and I/O.

3.1. UI requirements

In this section, the requirements regarding the system's user interface (UI), i.e., the human-computer interface, are specified.

No.	Description	Relevant user requirements
1.	The service shall provide adequate switching/selecting options for different input and output modalities.	U35
2.	The interface shall include feedback information in a user-friendly and adaptive way.	U36
3.	The date of the last map update shall be shown to the user if available.	U37
4.	All screens shall include a prominently placed help link.	U38

3.2. Functional requirements

In this section, the requirements regarding the underlying system functionality are specified.

No.	Description	Relevant user requirements
1.	The service shall only allow publication of help content if it is relevant for the problem to solve.	U18
2.	The service shall promote the conciseness of help content.	U18
3.	The service shall support help content split into several steps/segments and promote segmented content.	U18
4.	The service shall support the following content modalities: • Video • Still images • Audio/speech • Text/captions	U19a,b,c
5.	The service shall measure the accessibility of content and encourage the generation of accessible content.	U19

6.	The service shall support multiple content languages.	U20
7.	The service shall allow content to be associated with a unique tag/ location.	U21a, b, e
8.	The service shall provide clients requesting information on a particular location point with a corresponding map.	U22
9.	The service shall support the provision of navigation information in terms of: • departure times • arrival times • transportation means • travel prices and discounts	U22, U23a,b,c,d
10.	There shall be a user profile for customization purposes.	U24
11.	The user shall be able to customize the preferred UI language and the preferred content languages. The following UI languages shall be available: English, Romanian, Norwegian, and Spanish. The default UI language should be set according to the user's IP address. The content language shall not be limited and shall have the default value Automatic (same as UI language).	U24c, U32
12.	The service shall have 0 stability issues in ideal conditions (enough memory space, no other applications running in background, etc.).	U25
13.	The implementation shall follow WCAG 2.0 AA.	U27
14.	The service shall hide unwanted or not needed functionality from the user.	U28, U29
15.	In selection situations, the service shall give the user a reasonably limited number of options.	U30
16.	All map interfaces shall have the option to zoom in and zoom out.	U31
17.	The user profile shall contain service settings and shall be available for change at any time.	U34
18.	The service shall provide feedback on most user initiated actions and always on critical service functionality.	U36
19.	The service shall provide adequate and accessible help content for all of its functionality.	U39
20.	Any error, warning, and other messages shall be relevant and concise.	U40
21.	There shall be fallbacks for possible service failures.	U41
22.	The user profile shall be stored in a format that cannot be read by third parties.	U44
23.	It shall be possible to reset the user profile and to delete a user account.	U44
24.	The service shall show no content rather than irrelevant content.	U1
25.	The service shall allow to combine different modalities for the same help content, and for several help resources.	

3.3. I/O requirements

In this section, the requirements regarding the system interface in terms of input/output (I/O) operations are specified.

No.	Description	Relevant user requirements
1.	The service shall send media in formats that are compatible with the capabilities of the HoD client, including container formats.	
2.	The service shall accept as many multimedia formats as possible and be able to convert them in order to be understood by HoD. •	
3.	The service shall store the following video resolutions: • Original • Mobile (800x480 pixels)	
4.	The service shall store the following audio resolutions: • Original • AAC 128 Kbps	

5 Non-honored user requirements

These are the user requirements that have been ignored during the derivation of system requirements.

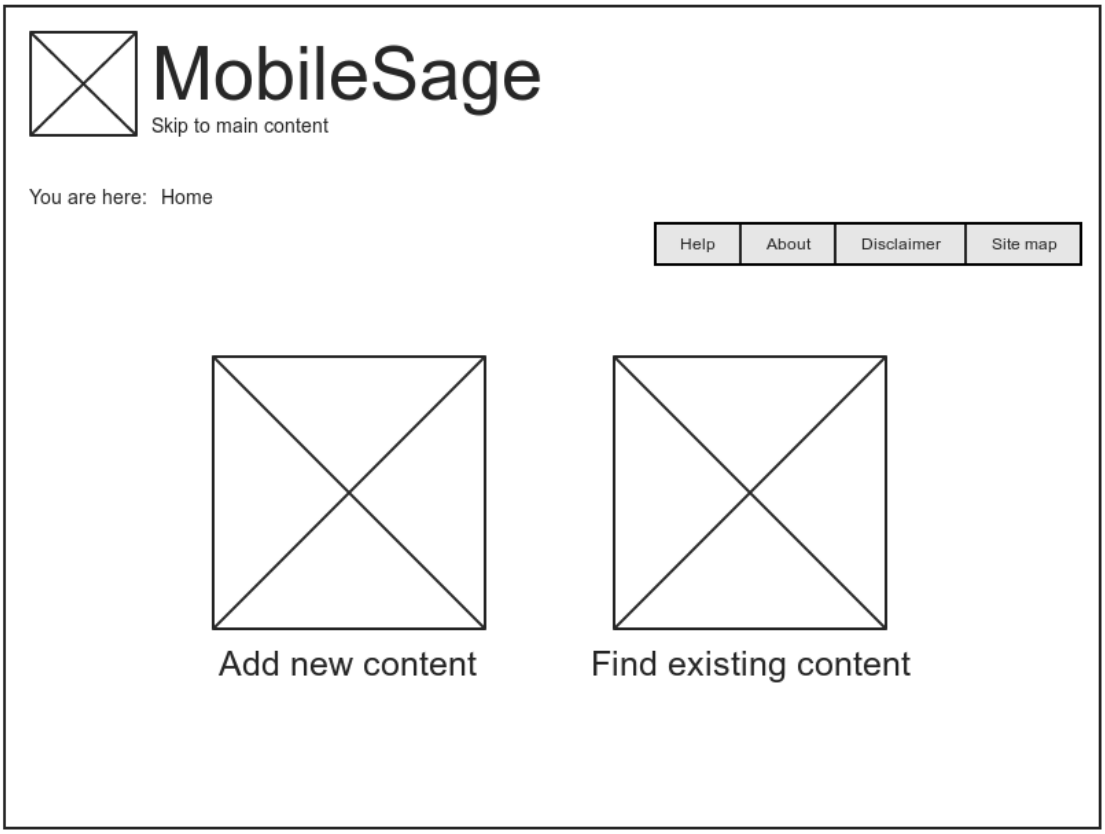
No.	Description	Relevant user requirements
1.	CM shall provide content on the following topics: • Cooking recipes • Rules for sports and games	U21c,d
2.	It shall be possible to customize CM with regard to functionality.	U24a
3.	It shall be possible to customize CM with regard to subservice used.	U24d
4.	The service shall motivate the user to help herself and acknowledge any self-managed accomplishments.	U42
5.	The service shall consider the use of avatars.	U43

6 Mockups

This section is informative only. The mockups should be treated as examples of how the user interface may look like.

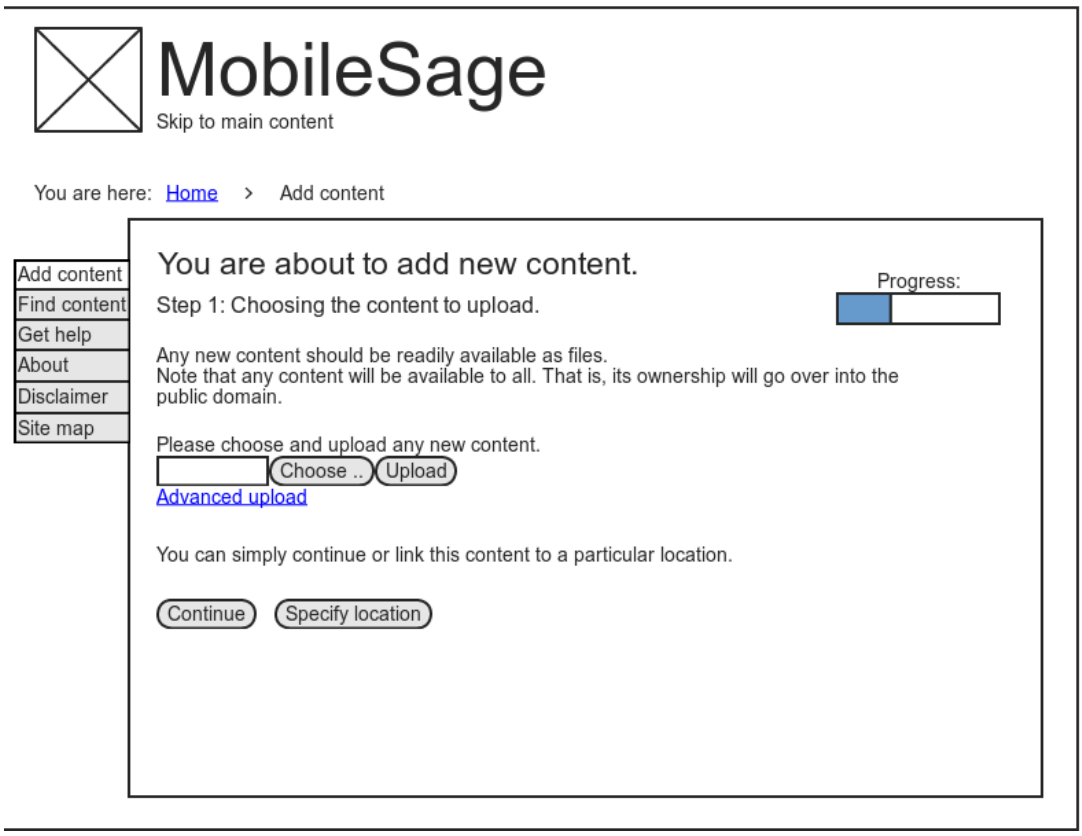
6.1 Front page

This mockup shows an example front page for Content Management.



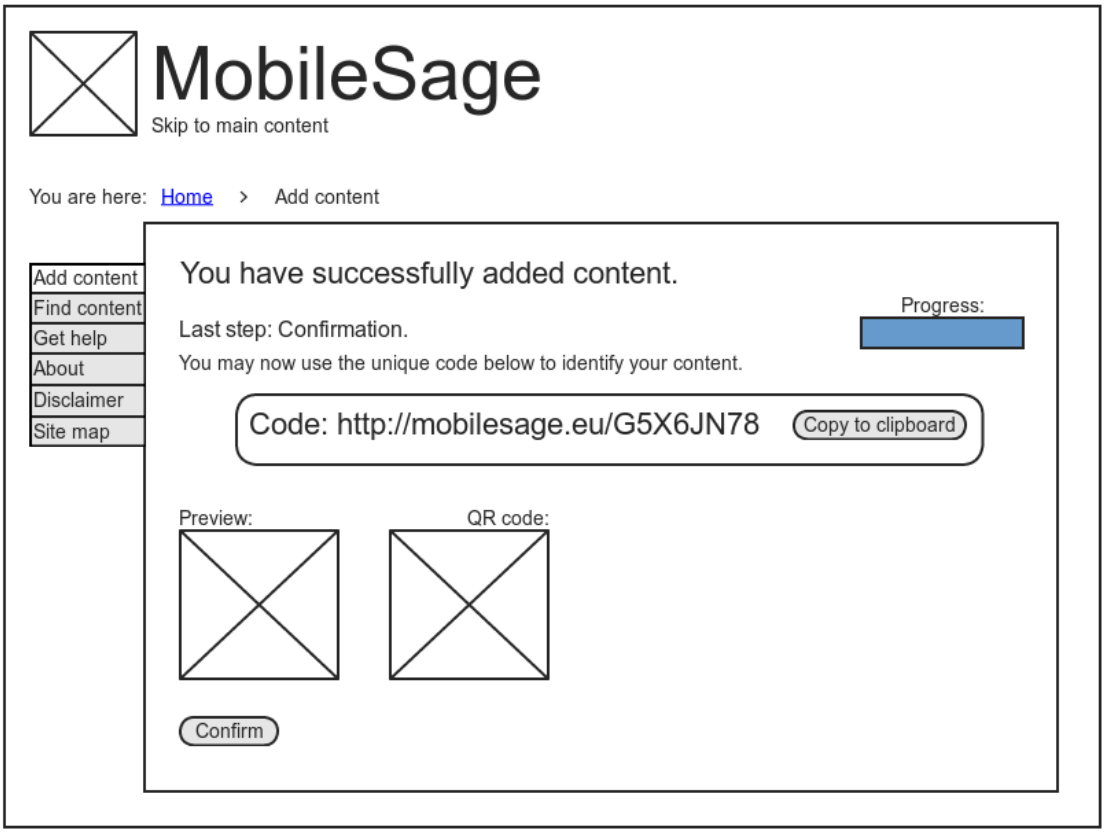
6.2 Add content: Choosing and uploading content

This mockup shows an example of specifying new content.



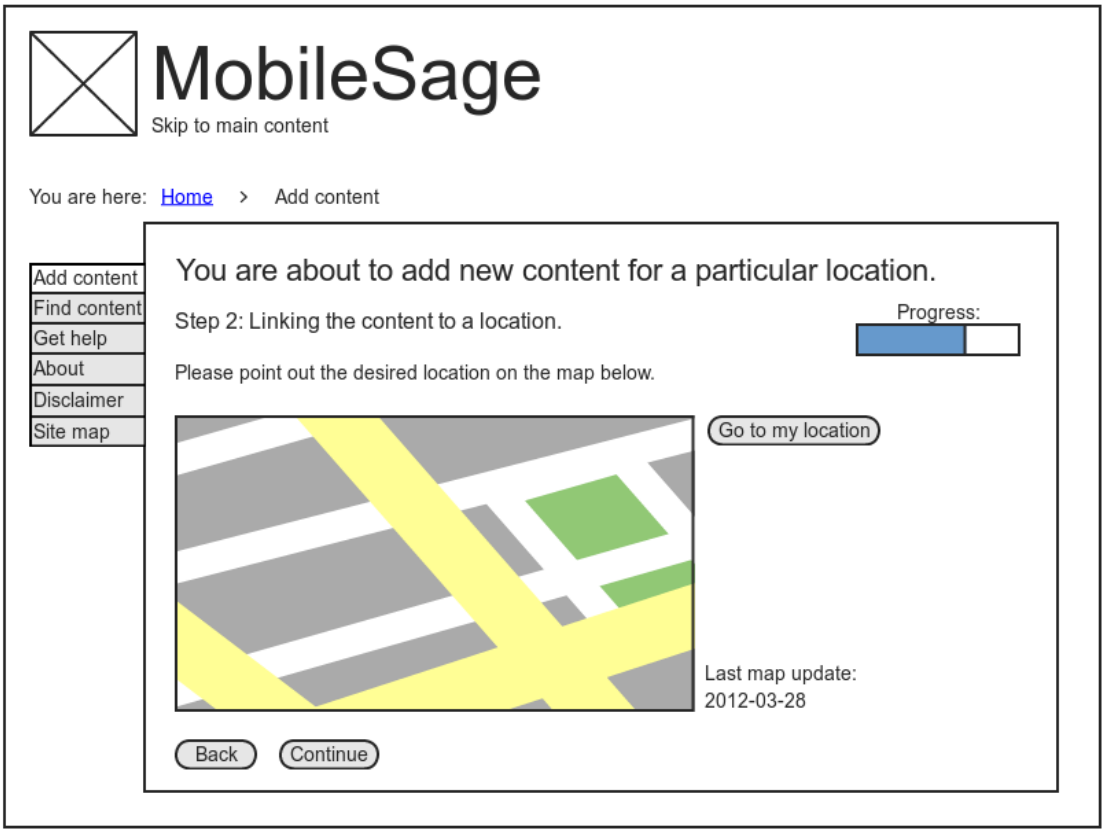
6.3 Adding content: Confirmation

This mockup shows an example of a confirmation screen for uploaded content.



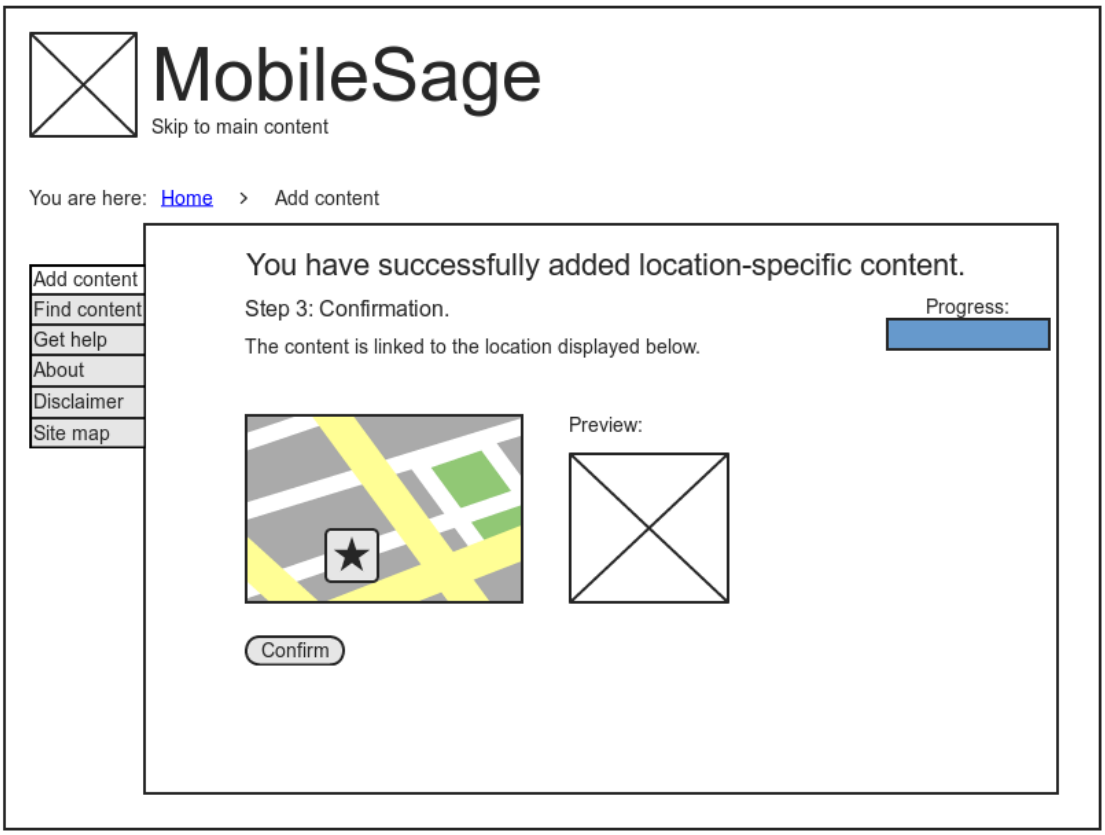
6.4 Adding location-specific content

This mockup shows an example of specifying a location for uploaded content.



6.5 Adding location-specific content: Confirmation

This mockup shows an example of a confirmation screen for location-specific content.



References

1 AAL Association in ((), 2010).
2 Sánchez, V. in ((), 2011).
3. Røssvoll, T.H. MobileSage Deliverable D2.2: User Requirements Specification. , (2012).
4. Curescu, L., Anghelache, I. & Røssvoll, T.H. D2.3 System Requirements Specification for Help on Demand Service. , (2012).