Functional specification
GDACSmobile

Project Seminar Global Disaster Alert and Coordination System (GDACS) mobile

Prof. Dr.-Ing. Bernd Hellingrath, Dipl.-Logist. Daniel Link, Adam Widera, M. A.

Carsten Bubbich, Friedrich Chasin, Sven Kronimus, Stefan Laube, Hendrik Petersen, Philipp Saalmann, Martin Vanauer

10.05.2012
## Content

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>3</td>
</tr>
<tr>
<td>Project description</td>
<td>3</td>
</tr>
<tr>
<td>Target groups</td>
<td>3</td>
</tr>
<tr>
<td>Core functionalities</td>
<td>3</td>
</tr>
<tr>
<td>Product usage</td>
<td>5</td>
</tr>
<tr>
<td>Interface design (GUI)</td>
<td>6</td>
</tr>
<tr>
<td>Functionalities</td>
<td>8</td>
</tr>
<tr>
<td>Client</td>
<td>8</td>
</tr>
<tr>
<td>Needs Assessment</td>
<td>8</td>
</tr>
<tr>
<td>Situation Overview</td>
<td>12</td>
</tr>
<tr>
<td>Global map overview</td>
<td>13</td>
</tr>
<tr>
<td>Local view</td>
<td>15</td>
</tr>
<tr>
<td>Server</td>
<td>18</td>
</tr>
<tr>
<td>Short message analysis</td>
<td>18</td>
</tr>
<tr>
<td>Role Management</td>
<td>21</td>
</tr>
<tr>
<td>Content Management</td>
<td>21</td>
</tr>
<tr>
<td>Application Architecture</td>
<td>22</td>
</tr>
<tr>
<td>General Architecture</td>
<td>22</td>
</tr>
<tr>
<td>Target Devices</td>
<td>22</td>
</tr>
<tr>
<td>Server Environment</td>
<td>23</td>
</tr>
<tr>
<td>Server Architecture</td>
<td>23</td>
</tr>
<tr>
<td>Interfaces</td>
<td>25</td>
</tr>
<tr>
<td>Project Plan</td>
<td>26</td>
</tr>
<tr>
<td>Inception Phase</td>
<td>26</td>
</tr>
<tr>
<td>Elaboration Phase</td>
<td>26</td>
</tr>
<tr>
<td>Construction Phase</td>
<td>26</td>
</tr>
<tr>
<td>Release 1: 11&lt;sup&gt;th&lt;/sup&gt; June</td>
<td>26</td>
</tr>
<tr>
<td>Release 2: 2&lt;sup&gt;nd&lt;/sup&gt; July</td>
<td>26</td>
</tr>
<tr>
<td>Final Release: End of July</td>
<td>27</td>
</tr>
<tr>
<td>Transition Phase</td>
<td>27</td>
</tr>
<tr>
<td>Work packages</td>
<td>29</td>
</tr>
<tr>
<td>Authors</td>
<td>32</td>
</tr>
<tr>
<td>Appendix A: Twitter Processing Details</td>
<td>33</td>
</tr>
</tbody>
</table>
Target

Project description
The GDACSmobile project is a cooperation of the Information Systems Department of the University of Münster and the Global Disaster Alert and Coordination System (GDACS) set by the Joint Research Centre (JRC) of the European Commission. The goal of the project is to create a mobile application (collectively referred to as "GDACSmobile") for information exchange and coordination support in the first phase after a major sudden-onset disaster.

To accomplish this goal, GDACSmobile will be implemented for a selected number of mobile platforms. The client will be supported by a server application which is responsible for data retrieval, storage and analysis while the client is primarily used for accessing content and sharing/updating information. In this context, the development team is mainly compromised of project seminar participants at the Chair for Information Systems and Supply Chain Management (University of Münster) while professional support concerning requirements, preconditions and restrictions is provided by the JRC.

Target groups
GDACSmobile aims at supporting two main target groups: people concerned with disaster relief, and the (affected) population itself. Both groups will be able to use the application for sharing information, thus in turn creating a better situational awareness which is crucial for effective disaster response.

Although both groups will be able to use the application, different rights and roles are assigned to the users. People concerned with disaster relief will be referred to as authorized persons. Those authorized persons are assumed to be working for a professional humanitarian organization (governmental or non-governmental) providing professional response services to the local population. Information retrieved from those users is believed to be highly accurate due to the professional background and thus are classified as authorized and trustworthy.

The affected population will also be able to use the GDACSmobile application, but is provided with comparably less functionality. Primarily, not all information which is relevant for professional disaster relief operations is directly needed by the affected population. Although the application will try to enable the local population to start self-organization, the primary focus lies on information retrieval to obtain a better awareness of the current situation. Consequently, the population will be provided with a limited ability to assess the current needs situation and submit this data to the server and thus to professional helpers receiving the information. A more detailed description of roles and associated rights is provided in the functionality overview (cf. Core functionalities, p. 3) and the product usage section (cf. Product usage, p. 5).

Core functionalities
The core functionalities of the GDACSmobile client consist of:

1. Differentiated visualization of information gained from application-based need assessment forms and short message services (e.g. Twitter). The usage of geo-information for visualization purposes will be an important part of the application. By providing geographic
maps and information concerning needs assessment, the disaster responders will be able to get a situation overview before accessing the disaster on-site.

2. Needs assessment functionality to provide an interface representing specific sectors of humanitarian response (Health, Infrastructure, etc.), such that a categorization and communication of field observations is possible.

3. Integration of social media crowd-sourced information by using short-message analysis. Based on the idea that the population affected by the disaster is using short-messages by means of e.g. Twitter/SMS for information dissimilation to the responders, those short-messages transferred to the implemented server will enrich the information assessed.

Summary:

- User data
  - Each user has a unique combination of username and password assigned
  - Each username is associated with a role
  - Access without username is possible (“unregistered”)

- User interface
  - The user interface is optimized for touchscreen smartphones
  - The functionality will be completely available for both finger and stylus input
  - The user interface is based on common mobile device design guidelines for highest usability and comprehensibility

- Needs assessment
  - Observation from the field can be categorized concerning major sectors of humanitarian response
  - Disaster responders can write plaintext in order to describe their observations quantitatively and qualitatively
  - Taking and attaching photos to specific observations is facilitated
  - Symbolic icons are provided to categorize the impact of observations. Moreover, the icons can be used to enrich the plaintext describing an observation
  - Tagging observations with geographic information is provided

- Situation overview
  - Assessed observations from the field are selectable over lists, corresponding to the major sectors of humanitarian response. Consequently, observations are categorized and it is possible to request information to specific humanitarian sectors only
  - Details to observed situations can include plaintext, photos, symbolic icons as well as geographic positions on maps

- Global map overview
  - Potential disasters world-wide are visualized on a map for disaster managers
  - Summaries of the disasters are provided

- Local view
  - Detailed map views of the areas affected by disasters are shown
  - Analysis results from needs assessment and short message analysis are visualized
  - Selection of map layers for different types, sources and trustworthiness of information
  - Time-dependent visualization of information
- Short message analysis
  o Short messages are automatically retrieved from selected sources
  o Messages are filtered according to defined filter algorithm (e.g. hashtag filter)
  o Web interfaces allows crowd analysts and team managers to manually screen messages and select relevant ones
  o Relevant messages are treated as “needs assessment” reports and can be enriched with further data (e.g. metadata like location)
  o Relevant messages must be manually organized according to disaster needs - specific sectors of humanitarian response (recommendation may be provided by algorithms)

- Server
  o Interfaces to allow a communication between server and client applications will be implemented
  o Situation reports will be provided using the interfaces
  o The server enables user role management

- Miscellaneous
  o English is the language of choice for the user interface and documentation
  o The final release will include an installation/deployment guide and a user handbook.
  o Functionalities of the software, which are available to the users, depend on their roles and rights
  o Each provided functionality is assigned to at least one role
  o The messages exchanged between the mobile and server application will be as small as possible to keep the network traffic low

**Product usage**

As already mentioned, GDACSmobile has two main target groups: authorized and registered people from humanitarian organizations and affected people. While the affected people are homogenous in terms of user rights and roles, the roles of the authorized people are different in terms of capabilities and rights regarding to the usage of the application. These rights grant or deny access to different functionalities of the application. An overview of the different roles and corresponding usable functionalities is given by Table 1.

<table>
<thead>
<tr>
<th>Mobile client</th>
<th>Not registered</th>
<th>Registered</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Population</td>
<td>First Responder</td>
</tr>
<tr>
<td>Needs Assessment</td>
<td>x (untrusted)</td>
<td>x</td>
</tr>
<tr>
<td>Situation Overview</td>
<td>x (filtered)</td>
<td>x</td>
</tr>
<tr>
<td>Global map overview</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Local View</td>
<td>x (filtered)</td>
<td>x</td>
</tr>
<tr>
<td>Short Message Analysis</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Role Management</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Content Management</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

*Table 1: User roles and associated functions*
- The *population* has access to the application and does not need to register. This means that everyone is able to download and use the application. Reports or needs assessments sent by this user role will be tagged as untrusted because the source and reliability of the information is unknown. The functionalities *situation overview* and *local view* are restricted, such that the population has limited access to information.

- The *first responders* as well as the *disaster manager* have to be registered, which means that these users will be tagged as trusted people. To prevent unnecessary data traffic, the first responder does not have a global map overview.

- The *disaster manager* is the super user of the mobile application.

- While the fore-mentioned roles are using the mobile application, the *crowd analysts* using a web application. His duties are the analysis and assessment of short messages.

- The *moderator* typically belongs to the organization maintaining the GDACSmobile client and GDACSmobile server. Users with this role will be able to add (or delete) users, assign roles to other users and furthermore moderate content (e.g. discard reports).

The registered user has to authenticate themself by a combination of username and password. If the device is connected to the internet, the combination will be verified on the server by a secured login request. This is mandatory for the initial login. If the combination of password and username is valid, the user has granted and role-dependent access to the functionalities of the application. If the login is not accepted, an error message pop ups and the user can try to login again. After three unsuccessful attempts, the user application will be blocked on the device and has to be unblocked from the server.

After the first login, the username will be stored on the device, such that a login without internet access is possible. In addition, the application provides an auto-login functionality so that the user does not have re-login every time he wants to use the application.

Since disasters can have a serious impact on a country's infrastructure, it might be possible that mobile internet access is available to a limited extent. While some functionality will not work properly without internet access, the functions *needs assessment* and *situation overview* are able to work without an established network connection (i.e. the request of concerning information - no response functionality). Table 2 shows which functionalities are provided in the offline mode on the mobile client.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Offline access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Needs Assessment</td>
<td>x</td>
</tr>
<tr>
<td>Situation Overview</td>
<td>x</td>
</tr>
<tr>
<td>Global map overview</td>
<td>-</td>
</tr>
<tr>
<td>Local View</td>
<td>-</td>
</tr>
<tr>
<td>Short Message Analysis</td>
<td>-</td>
</tr>
</tbody>
</table>

*Table 2: Functionality provided without internet connection*

**Interface design (GUI)**

The goal of the application’s GUI-concept is to ensure an easy and comprehensible usage within crisis environments. As a result, the user can access all functionalities while paying minimal attention to the application in order to have maximal attention for his or her environment. This will be done taking the following guidelines into account:
- The interface will be as simple as possible without overwhelming the user with too many options
- The amount of the displayed information can be changed through dynamic constraint filter and personalization options
- The user will be guided through necessary input steps in a very compact way. Moreover, as many information will be gathered automatically to avoid unnecessary input steps
- The application’s design is consistent within all menus.

Most of the automatically gathered information is dependent on context-awareness functionality, e.g. current geo-related position. The application’s design is optimized for being used on touchscreen devices. Nevertheless, some functionality will still be usable with keypad-based input.

The mockup shown below (Figure 1) visualizes the startup screen of GDACSmobile and consists of buttons guiding to the main functions of the application. The buttons might be partially disabled, according to the user’s role. The button on the upper left leads directly to the first need assessment menu. The button in the lower left opens to the global map and provides information about ongoing disaster events. Both buttons on the right side are referring to the currently set default disaster event. While clicking the upper one a detailed map view is displayed. Using the lower button the situation overview report will be visible. The concrete functionalities accessed through the main screen of the GDACSmobile client will be described in detail in the following section. The visualization mockups in those sections are used for illustrative purposes and do not necessarily reflect the GUI of the final GDACSmobile client.

![Figure 1: Start screen of GDACSmobile client](image-url)
Functionalities

Client

Needs Assessment

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Needs Assessment</td>
<td>The needs assessment functionality will provide an interface representing specific sectors of humanitarian response (Health, Infrastructure, etc.) such that a categorization and communication of field observations is possible.</td>
</tr>
<tr>
<td>Prerequisites</td>
<td>A prerequisite for the submission of needs assessed with this functionality is an established connection to the internet. Reports are however cached for a limited time during offline mode and transferred when the server can be reached again.</td>
</tr>
<tr>
<td>Access</td>
<td>This functionality will be available for everyone having the application installed. There will be no differentiation between the stakeholders. However, authorized humanitarian responders will be considered more trustworthy than other people assessing needs.</td>
</tr>
</tbody>
</table>

Application process for assessing needs: The needs assessment sector will be available via one click. After choosing needs assessment, the user will be provided with a list of ten different major sectors of humanitarian response (Major sectors interface). These sectors are based on the Sphere Project and OCHA handbook and include “Water and Sanitation”, “Shelter and Settlement”, “Health”, “Infrastructure”, “Agriculture”, “Food”, “Nature of Disaster”, “Logistics”, “Search and Rescue (SAR)”, and “Nutrition”. Every sector will be represented with a different icon, and all icons have a specific sector name annotated. At this point, the person assessing needs will have the possibility to select a specific icon of major humanitarian response in order to categorize an observation. After selecting an icon, the application will lead to a specific subsector of humanitarian response (Subsector interface), depending on the major sector chosen. The person assessing needs will have to choose between the different subsectors in order to input data. The different major sectors with their pertaining subsectors are represented in Figure 2.
Selecting a subsector of humanitarian response leads to an input interface. This interface will provide the possibility to input plaintext, take (or load) a picture, store geographic information, and select between different symbols highlighting the importance of the observation.

- **Input plaintext:** The plaintext input field can be used for qualitative and quantitative needs assessment (Input field plaintext). The person assessing needs will have the possibility to write a short text (up to 300 characters) summarizing the observation.

- **Take a picture:** This input icon will lead to the camera functionality of the mobile device (Icon “Picture”). After a picture is taken, the person assessing needs will have the possibility to store the picture or try again. The application will return to the input interface just after a picture is stored or the picture-taking process is aborted. Since there will be the possibility to take one picture only, the last taken and stored picture will be available in the end. There will also be the possibility to load a picture out of the local picture library (Icon “Load Picture”).

- **Store geographic information:** There will be an icon for storing the current position in the field (Icon “By GPS”). Moreover, the responder has the possibility to click on an icon to select a specific geographic position on a map (Icon “By map”).

- **Select between different symbols:** The person assessing needs will have the possibility to select between different intuitive symbols to categorize the effect of his observation to the mission. The different symbols for every subsector will be evaluated and selected during the implementation phase. (Symbols “List of symbols”)

Finally, the user of the application will be able to send the assessed data to the server (Icon “Submit”). Moreover, regardless whether there is an internet connection available, the information
will be stored on the mobile device itself. In case that information could not be send due to the lack of an internet connection, the application will try again as soon as connectivity is provided.

*Use case: Observing a destroyed bridge*

1. **Application interface:** The humanitarian responder has to select the implemented needs assessment functionality.
2. **Major sectors interface:** A destroyed bridge concerns the infrastructure of the disaster area. Consequently, the humanitarian responder has to select the “Infrastructure” sector of humanitarian response.

![Figure 3: Needs assessment sector selection](image)

3. **Subsector interface:** The subsector interface concerning the “Infrastructure” sector will consist of five different items, namely: “Communications”, “Electric Power”, “Water/Sewage”, “Hydro Facilities”, and “Roads and Bridges”. From these items, the subsector “Roads and Bridges” has to be selected in order to categorize the observation.
4. Input interface: The responder has the possibility to insert plaintext like “Destroyed bridge; Not usable” (Input field plaintext). In order to confirm the observation, it is wise to take a picture (using the take a picture functionality) of the destroyed bridge (Icon “Picture”). At this point, the responder should also save some geographic information related to his observation. Therefore, he can either store his current position (Icon “By GPS”) in the field (in case he is close to the bridge) or store the actual geographic position of the bridge by selecting it on a map (Icon “By map”). To highlight the importance of the observation, the responder can select the “Red” symbol of the list of symbols. (Symbols “List of symbols”) This illustrates, that the observation can have major effect on the humanitarian response itself.
Situation Overview

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The situation overview functionality is implemented to give disaster responders, registered on the server, a better awareness of the current disaster situation. Additionally, people of the population will have the possibility to access a filtered situation overview.</td>
<td></td>
</tr>
</tbody>
</table>

| Prerequisites | A prerequisite for this functionality is an established connection to the internet. If there is no connection to the internet possible, a locally stored situation overview will be loaded. This stored situation overview represents the disaster situation from the time the last internet connection has been established. |

| Access | This functionality will be available for everyone having the application installed. There will be a differentiation between registered humanitarian responders and other stakeholder. While registered first responder and disaster manager will have full access to the information of the situation overview, information for the population will be filtered. |

**Application process for accessing the situation overview:** The situation overview (SO) sector will be available via one click. After choosing situation overview, the user will be provided with a list of ten different major sectors of humanitarian response (Major sectors for the SO interface). These sectors are based on the Sphere Project and OCHA handbook and include “Water and Sanitation”, “Shelter and Settlement”, “Health”, “Infrastructure”, “Agriculture”, “Food”, “Nature of Disaster”, “Logistics”, “Search and Rescue (SAR)”, and “Nutrition”. Every sector will be represented with a different icon, and all icons have a specific sector name annotated. At this point, the person interested in the situation overview will have the possibility to select a specific icon of major humanitarian response in order to get information concerning this topic. After selecting an icon, the application will lead to a list of needs which already have been assessed concerning the topic (Situation overview list concerning one topic). This list provides access to specific needs assessment reports. The structure of each specific report (Specific report interface) is depending on the information assessed. A report can consist of a picture, plaintext, and specific symbols – all depending on the information assessed. Moreover, stored geographic positions concerning the report are available via an icon (Icon "Go to map").

**Use case: Infrastructure Situation Overview**

1. Application interface: The stakeholder has to select the implemented situation overview functionality.
2. Major sectors for the SO interface: The stakeholder has to select the “Infrastructure” sector of humanitarian response.
3. Situation overview list concerning one topic: The stakeholder has the possibility to choose between different reports which have been assessed concerning the major sector of humanitarian response. If he wants to have a look at the destroyed bridge (Cf. needs assessment use case: Observing a destroyed bridge), he clicks on “Destroyed bridge; Not usable”.
4. Specific report interface: This interface consists of the information assessed. If there is geographic information stored for the specific report, an icon with a link into the local view map interface will be accessible (Icon "Go to map").
**Global map overview**

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>An overview about potential disasters world-wide is shown on a map. The information about these events is provided by GDACS.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prerequisites</th>
<th>Established internet connection, access to GDACS API for alerts.</th>
</tr>
</thead>
</table>

| Access | This functionality will be available to disaster managers. |

**Application process:** The GDACSmobile application uses a map in order to visualize potential disasters world-wide. Map data is retrieved from selected web map services. Iconic symbols represent the information on the map at the corresponding geographic positions. Both, the type of disaster and its impact are shown by the standard GDACS icons. The mockup Figure 7 illustrates the map projection of GDACS alerts. The zooming and panning controls on the left hand side facilitate map navigation. The upper four controls move the map towards the four cardinal directions. The lower controls enable users to zoom in or out.

![Global view map with potential disasters displayed](image)
Use case: Request additional information

The disaster manager clicks on a disaster icon in order to open a tooltip with a short description of the disaster (see mockup Figure 8).

![Figure 8: Selection of disaster](image)

Regarding the tooltip, the button “more info” opens a summary view of the disaster with additional information (see mockup Figure 9). Both buttons “Detailed Map” in the tooltip as well as in the summary view forward the disaster manager to the local view of the affected area which is presented in the following section. Afterwards, the view for general information corresponding to the disaster is presented. This view is accessible via the button “Statistics”.

![Figure 9: Detailed information retrieved from GDACS for disaster](image)
**Local view**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functionality</strong></td>
</tr>
<tr>
<td><strong>Prerequisites</strong></td>
</tr>
<tr>
<td><strong>Access</strong></td>
</tr>
</tbody>
</table>

*Access application process:* The information related to the selected disaster is shown as icons on the map. The icons vary in shape, size and color for the different types and sources of information. Depending on the zoom level, icons, which are geographically close to each other, are clustered and visualized by a different icon in order to show the grouping. Thereby, overlapped and hidden icons are avoided. For instance, mockup Figure 10 illustrates the cluster of two water supplies.

![Figure 10: Clustered view of 2 alerts/situation reports](image)

The different zoom levels provide users with more detailed information. The clusters which are generated in order to limit the amount of icons could be further aggregated or split into smaller clusters depending on the zoom level. Mockup Figure 11 shows exemplarily the split up of the water supply cluster from mockup Figure 10 for a more detailed zoom level.

![Figure 11: Detailed information visualization](image)
Icons are allocated to separate map layers. A map layer stores coherently a set of graphical information and could be displayed as an overlay on the map. The layers could be combined simultaneously on the map and are pre-selected for the different user roles. Users could customize the view using the menu on the right hand side (see mockup Figure 12).

![Figure 12: Customization of local view](image)

**Use case: Select information layers**

The menu could be opened by clicking or dragging the menu drag bar (see mockup Figure 13). The menu shows the four menu items “Categories” (e.g. needs, specific situation reports), “Trustworthiness”, “Sources” and “Time” in order to adjust the selection of displayed information.

![Figure 13: Map filter - categories](image)

A click on the menu item “Categories” opens a select list showing the ten major sectors from needs assessment (see mockup Figure 14). Categories could be chosen by selecting the corresponding check boxes and the related information are displayed on the map.
After clicking on the item “Trustworthiness”, users could choose to display verified and/or unverified information (see mockup Figure 15). Users which are registered and thus able to submit verified reports will also be able to select unverified information and mark them as verified.

Different information sources could be selected via the menu item “Sources”. Mockup Figure 16 illustrates the selected sources GDACS and Twitter.
The menu item “Time” enables users to define time spans for time-dependent information. The start time and end time could be picked as illustrated in mockup Figure 17.

![Figure 17: Map filter - time](image)

### Server

#### Short message analysis

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functionality</strong></td>
</tr>
<tr>
<td><strong>Prerequisites</strong></td>
</tr>
<tr>
<td><strong>Access</strong></td>
</tr>
</tbody>
</table>

*Application process:* The GDACSmobile application integrates short message services like Twitter as a source of information. Such services here serve as a mechanism to quickly identify trends, needs of the affected population and other important aspects relevant for disaster response. Information retrieved can for example include local calls for help or information concerning the unavailability of certain infrastructure.
For the purpose of information retrieval, in the case of Twitter, trusted channels are crawled for disaster-relevant information. Other identification methods could include but are not limited to the definition of specific hashtag patterns which allows the affected population and other participants to share information like the current position of disaster events. In the case of wildfires for example, users would be able to share which streets are currently unusable due to fire - providing first responders with valuable information which would not be available otherwise and thus significantly delay rescue operations. Those disaster-related tweets of affected population are recognized by using specific filter mechanisms. (As outlined e.g. user defined hashtag combinations, algorithmic classification; final implementation decision depends on feasibility analysis performed during initial architecture design, cf. Figure 18)

Data retrieved by short message services is not directly accessible from the client but first enriched and analyzed. It is critical to note that all data processing is semi-automatic and will be performed on a server crawling short message data. In addition to obvious performance reasons, this also enables the integration of crowd-sourcing for data retrieval. For example, a limited group of users will manually review selected tweets and enrich them with further information which in turn are then transmitted to the responders. For this purpose, messages which were identified as potentially relevant are displayed to volunteers on a webpage. On this webpage, those so called Crowd Analysts will be able to quickly scan the identified tweets and enrich relevant tweets with further information. This can include for example setting geolocations, adding pictures etc. The interface will be based on the Needs Assessment functionality and offers the same input options. The data then is made accessible to the first responder in the local view, similar to local situation reports.

Twitter processing is described in detail in Appendix A.
Use case: Reviewing tweets about a disaster - adding situation report to local view

After a major earthquake, Twitter users quickly begin to adapt the hashtag #zzz to share information about the current situation in the affected area. Through automatic analysis, only tweets which match this hashtag are displayed on a webpage to Crowd Analysts in order to allow manual filtering and enriching the data.

![Figure 19: Web interface for crowd collaboration](image1.png)

Crowd Analyst x reads a tweet #zzz Bridge on Suffolk Avenue destroyed. He quickly recognizes that this information should be made available to first responders and starts searching for coordinates of the bridge. After identifying the affected bridge, he uses the web interface to select the appropriate category (Infrastructure), sub category (Roads and bridges) and sets the location identified. After submitting this report, it is automatically displayed on local/first responders’ local view with the appropriate icons.

![Figure 20: Short message acceptance for situation report](image2.png)
Role Management

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moderators are able to add and remove users as well as assign roles to existing users.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prerequisites</th>
<th>None.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>This functionality will be available to moderators.</td>
</tr>
</tbody>
</table>

As outlined in the section Product usage, the GDACSmobile client is used by a variety of users. Users are assigned to different roles in order to allow usage of the client as well as the server application. These roles grant or deny access to different functionalities of the application. On the server, users with the assigned role moderator are able to create, block, unblock and delete users. For this purpose, a simple administrative user interface (UI) is offered.

Furthermore, through the same UI, roles can be assigned to individual users. By default, new users are not assigned to any role and thus are basically only able to use the mobile application as untrusted users. By assigning them individual roles (e.g. first responder or disaster manager) they are marked as trustworthy and get access to more information.

Content Management

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submitted reports are not displayed on the mobile devices per se but can be accepted or deleted. Furthermore, even already accepted messages can be deleted afterwards if deemed necessary.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prerequisites</th>
<th>None.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>This functionality will be available to moderators.</td>
</tr>
</tbody>
</table>

Content stored on the GDACSmobile server is mainly a consequence of reports received from various users using the GDACSmobile application. Depending on the user type (registered or unregistered), reports are assigned a trustworthy or untrustworthy flag. Reports coming from registered users for example are, given the professional background of registered users, treated as trustworthy.

The trustworthy flag is primarily responsible for distinction, i.e. which messages are automatically distributed to clients. Trusted messages are automatically accepted and thus distributed. Messages from non-trustworthy reporters are marked as non-accepted, meaning that they are not delivered to mobile clients via the interfaces. Those messages need to be accepted beforehand.

On the server, this is handled by a user interface which is similar to the short message analysis. For untrusted/not accepted messages, moderators are able to accept or decline reports. If a message is accepted, either by automatism or manually, it is accessible in a different interface where it can be deleted manually.
Application Architecture
The application which will be developed consists of two distinct parts, the mobile application itself and server-side components. The next section gives a brief overview of the general architecture of both parts and their interaction with each other. The brief overview will be followed by more details concerning the mobile devices, server environment and architecture.

General Architecture
The general setting of the application GDACSmobile is visualized in Figure 21. The GDACSmobile server application is the core of this framework dealing with tasks related to data retrieval, data storage and handling requests from the GDACSmobile application.

The mobile application provides functionalities which enable the user to access situation reports or send needs assessment to the server. Furthermore the visual representation of the crisis data is done via the mobile devices. Besides these two functionalities which have to be developed, interfaces of Web APIs like OpenStreetMap, Google Maps and Twitter will be used. This third party components either deliver content (e.g. Twitter) or provide mapping capabilities to our applications.

![Diagram of GDACSmobile environment](image)

Target Devices
The mobile application will mainly be developed to run on smartphones, i.e. the graphical user interface will be optimized for small screens. Supported operating systems will be iOS and Android. Furthermore the possibilities of cross platform development will be figured out, which might lead to limited functionality concerning some features. Depending on the technical feasibility and time constraints, an implementation for Windows Phone will also be considered. Necessary prerequisite to run the application and use all features is the availability of the following functionalities:

- Geolocation capabilities (e.g. GPS, WiFi and GSM/UMTS)
- Network connection for server communication (see offline access, cf. Product usage)
- Camera support
Server Environment

The server side components will be based on a comprehensive Java Enterprise Edition 6 (Java EE 6) architecture which will require a JEE 6 compliant application server. Due to the available expertise, functionality and associated costs, the JBoss Application Server 7.1.1 Final is selected as the primary application server. JBoss, which is primarily maintained by Red Hat, is a free of charge open-source product which is perceived as one of the leading enterprise application deployment platform servers, highlighting the high quality of the product. This is confirmed by the certification of standard compliance performed by Oracle.

In comparison to a simple Tomcat/JavaServlet platform, the use of a full Java EE 6 stack provides many significant advantages which are connected to the future maintenance of the GDACSmobile platform and to the speed of development. The following list provides a selected number of challenges to be faced if a simple configuration is used:

- Enterprise Java Beans (EJBs) are not available. This would lead to a not scalable business logic and can cause significant performance problems when the number of users respectively server requests increases. A Full-Profile compliant Java EE server takes care of these issues by instantiating and destroying EJB instances as they are needed.
- Web services based on SOAP or REST are not supported directly. Alternative implementations have to be found and tested.
- The Java Persistence API (JPA) is not natively provided within an Apache Tomcat server. JPA is an API providing an object-relational mapping (ORM), i.e. converting Java objects in database entries and vice versa.
- Java Server Faces (JSF) is not supported by Apache Tomcat. JSF is a further development of the Java Server Page (JSP) technology and decreases the implementation effort of web page considerably compared to JSP.
- Container services like dependency injection, transaction handling and security related issues are not natively provided by Apache Tomcat. Alternative implementations have to be found and tested.
- Portability advantages are lost since a Java EE compliant implementation can be run on every Java EE compliant application server.

A JBoss or more specific a Java EE 6 compliant implementation provides all capabilities to build a scalable and portable application, i.e. EJB, SOAP, REST, JPA and the other technologies are available out of the box.

Server Architecture

Figure 22 provides a more detailed overview of the server application and the connections to the related components. The server application consists of four different layers, namely data management, data, business and web tier. The data management tier is the persistent data storage of the data and consists of a database, e.g. MySQL. The remaining layers are part of a Java EE compliant implementation. A data model will be developed with respect to the different input sources and users. This model is part of the data tier and uses the Java Persistence API (JPA) which allows an automatic mapping of relational data from the database to Java objects. Consequently, an object oriented view is prevalent, eliminating the need to consider manual database design by creating tables. As a result, the technical specification will not be based on a traditional database scheme but on an object based view on the applications data needs.
The core of the server application is the business tier which encapsulates the business logic. The concept of Enterprise Java Beans (EJB) is used to provide a modular framework with components that allow an easy exchange of the actual implementation in order to deal with changing requirements. The main modules of the business logic will deal with the following topics:

- **GDACSmobile Connection Module**: Receiving needs assessment data and handling requests of the mobile application.
- **Crowd-sourcing Module**: Retrieving crowd data (e.g. twitter) and post-processing of these data.
- **Needs Assessment Module**: Maintenance of needs assessment data. This module also is responsible for retrieving data from external sources providing needs assessment reports. This could include – if technical feasibility permits – data stored on the iGDACS server accessed via interfaces.
- **Situation Report Module**: Storing and processing of data for situation report visualization.
- **Geo-Map Module**: Interface for a visualization based on maps. Most likely connections to Web APIs like Google Maps, OpenStreetMap will be used. Furthermore the usage of GeoServer will be evaluated.
- **Process Integration Module**: Integrates the previously described modules.

Finally the web tier will provide different modules dealing with e.g. the configuration of server side components like user management. Furthermore web pages based on Java Server Faces (JSF) provide views for tasks related to e.g. the categorizing and maintenance of tweets, or the situational overview of crisis.
Interfaces

In order to allow a communication between server and client application, other server application interfaces are necessary. These interfaces depend on the specific communication partners and will include SOAP-based web services as well as RESTful web services. Both interface implementations will communicate via the HTTP protocol. Furthermore the server side implementation makes use of the frameworks Java API for RESTful Web Services (JAX-RS) and Java API for XML Web Services (JAX-WS). The exchanged data packages will be optimized in order to decrease the network load as much as possible.

Due to the use of a full EE environment, typical features like automatic web service export will be used by the GDACSmobile server development team. As a result, web service interfaces are highly flexible and can be extended without major refactoring efforts. The following list provides an indication which services will be provided by the application. It is however not meant to be exhaustive as new services can be added continuously when application needs arise:

- #getAllReports(): returns a list of all reports with basic data
- #getFilteredReports (Criteria crit): returns a list of filtered reports according to criteria specification
- #getReport (int id): returns full details of a report, including picture data, author information etc.
- #submitReport (Category cat, String description, Image picture, Location geoPos): submits a new need assessment. Depending on preceding authentication, this is marked as accepted/non accepted by default
- #getGlobalReports(): returns a list of global disaster reports, most likely retrieved from the iGDACS server
- #login (String username, String password): returns an authentication token if successfully authenticated

External applications are easily able to access data which is stored on the GDACSmobile server. In order to further ease the use of the interfaces, an extensive technical documentation including certain tutorial steps will be provided by the development team.
**Project Plan**

The GDACSmobile project plan refers to the phases which are mentioned in the Rational Unified Process (RUP) framework. RUP structures software development projects in two the dimensions time and content. The four consecutive phases Inception, Elaboration, Construction and Transition are target-oriented. Furthermore, milestones must be defined on which stop or go decisions will be made.

The following paragraphs will provide an overview on the corresponding phases of GDACSmobile.

**Inception Phase**

The core functionalities of GDACSmobile, the mobile application and the application server, have been defined in the inception phase. Use cases were created in order to clarify the necessary functionalities. These use cases describe the interaction between actors and the application, and illustrate the definitions of the functionalities. The outcome of this phase is shown above in the different chapters about functionalities and application architecture.

**Elaboration Phase**

During this phase the project plan of GDACSmobile was constructed. For structuration purposes, different work packages have been created. For each package, the connected functionalities and their dependencies on each other were defined. The result is a project plan which is visualized by a GANTT-diagram in Figure 23.

**Construction Phase**

The implementation process is structured into three releases. Some of the defined functionalities are related to just one release while others must be enhanced during the whole implementation process. The latter are writing the installation and deployment guide, the preparation of the user handbook and software tests. This means that the guides and handbooks will be created for the functionalities simultaneously, while the continuous test package ensures a high software quality. Another ongoing work package is the development of the server architecture since it is a part of both the first and the second release. The remainder of this section contains the functionalities grouped by the three releases.

**Release 1: 11th June**

In the first release the functionalities

- needs assessment and
- situation overview

will be realized. The applications’ GUI and the server will be able to provide basic features in order to support the submission and the visualization of field observations.

**Release 2: 2nd July**

In the second release, the functionalities

- global map overview
- local view of the affected area
- user management
- server architecture foundation
will be included. Furthermore, the server will perform all service functions which are necessary to run the GDACSmobile application.

**Final Release: 16th July**
The final release will contain, among the aforementioned packages, the following functionalities:

- content moderation
- disaster selection
- mobile application foundation
- adoption to different platforms
- integration of iGDACS
- interfaces for data access
- short message analysis
- installation and deployment guide
- user handbook

**Transition Phase**
During this phase, the final product is completely developed and GDACSmobile will be handed over to the JRC. A software-field test will be part of this phase. The test will include an evaluation of the software to provide the JRC with an analysis of the software’s abilities and shortcomings. The concluding presentation is planned to be conducted at the end of July. The installation and deployment guide will offer further assistance while GDACSmobile will be deployed.
Figure 23: Gantt chart of the project plan
**Work packages**

The work packages with their priorities and efforts are provided in Table 3. The first column contains just the numeration of the work packages. Since some work packages and their functionalities consist of several sub-packages, the second column and the different row colors indicate the hierarchical level of the individual packages. The estimated efforts in man-days are displayed in second to last column. Hereby the reader has to take into account that a typical man-day of a project participant does not necessarily consists of eight working hours. The last column of the table shows the assigned priorities. These priorities show which work packages will absolutely be part of the final product (+) and which packages may, under unforeseen circumstances, not be part of it (O & -). Finally the higher level packages show the summed workload of their corresponding sub-packages.

<table>
<thead>
<tr>
<th>No.</th>
<th>Level</th>
<th>Name</th>
<th>Effort</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Functional Specification</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Application</td>
<td>276</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>Disaster selection</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>Based on GEO position</td>
<td>18</td>
<td>+</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>Manual</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>Mobile Application Foundation</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>Authentication</td>
<td>8</td>
<td>+</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>Permissions</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>Startup Screen</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>Adoption to different platforms</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>iOS</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>Android</td>
<td>10</td>
<td>+</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>Windows Phone 7</td>
<td>10</td>
<td>+</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>Needs Assessment</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>Major sectors interface</td>
<td>10</td>
<td>+</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>Different subsector interfaces</td>
<td>10</td>
<td>+</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>Input interface</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>Plaintext</td>
<td>5</td>
<td>+</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>Take Picture</td>
<td>3</td>
<td>+</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>Store geographic information</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>5</td>
<td>automatic</td>
<td>2</td>
<td>+</td>
</tr>
<tr>
<td>22</td>
<td>5</td>
<td>manually from map</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>4</td>
<td>Select Symbols</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>3</td>
<td>Data submission</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>Online</td>
<td>4</td>
<td>+</td>
</tr>
<tr>
<td>26</td>
<td>4</td>
<td>Offline</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>2</td>
<td>Situation Overview</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>3</td>
<td>Situation overview interface</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>3</td>
<td>Situation overview lists</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>Data retrieval</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>4</td>
<td>Online</td>
<td>8</td>
<td>+</td>
</tr>
<tr>
<td>32</td>
<td>4</td>
<td>Offline</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>2</td>
<td>Global Map Overview</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>3</td>
<td>Map integration (global &amp; local view)</td>
<td>25</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>3</td>
<td>Map tooltips</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>4</td>
<td>Summary view of Disaster</td>
<td>10 +</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>3</td>
<td>Data retrieval (online)</td>
<td>15 +</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>2</td>
<td>Local view of the affected area</td>
<td>51 +</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>3</td>
<td>Icons on the map</td>
<td>15 +</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>3</td>
<td>Clustering of icons</td>
<td>12 +</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>3</td>
<td>View customization (menu bar)</td>
<td>19 +</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>4</td>
<td>Menu bar 1: Categories</td>
<td>6 +</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>4</td>
<td>Menu bar 2: Trustability</td>
<td>3 +</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>4</td>
<td>Menu bar 3: Sources</td>
<td>5 +</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>4</td>
<td>Menu bar 4: Time</td>
<td>5 +</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>3</td>
<td>Data retrieval (online)</td>
<td>5 +</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>1</td>
<td>Server</td>
<td>80 +</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>2</td>
<td>User management</td>
<td>7 +</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>3</td>
<td>Add user</td>
<td>2 +</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>3</td>
<td>Delete user</td>
<td>1 +</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>3</td>
<td>Block user</td>
<td>1 +</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>3</td>
<td>Unblock user</td>
<td>1 +</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>3</td>
<td>Right assignment</td>
<td>3 +</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>2</td>
<td>Server Architecture Foundation</td>
<td>25 +</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>2</td>
<td>Short message analysis</td>
<td>28 +</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>3</td>
<td>Twitter</td>
<td>7 +</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>4</td>
<td>Retrieval of tweets</td>
<td>2 +</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>4</td>
<td>Filter mechanisms</td>
<td>5 +</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>3</td>
<td>SMS</td>
<td>7 +</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>3</td>
<td>Interface for crowd-sourcing (Web page)</td>
<td>14 +</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>4</td>
<td>Manuel filtering</td>
<td>3 +</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>4</td>
<td>Enriching data</td>
<td>11 +</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>5</td>
<td>Major sectors interface</td>
<td>3 +</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>5</td>
<td>Different subsector interfaces</td>
<td>3 +</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>5</td>
<td>Input interface</td>
<td>2 +</td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>6</td>
<td>Plaintext</td>
<td>1 +</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>6</td>
<td>Store geographic information</td>
<td>1 +</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>5</td>
<td>Data submission</td>
<td>3 +</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>2</td>
<td>Content moderation</td>
<td>18 +</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>3</td>
<td>Interface for moderator</td>
<td>18 +</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>4</td>
<td>Manual filtering</td>
<td>18 +</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>5</td>
<td>Authorized information</td>
<td>4 +</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>5</td>
<td>Unauthorized information</td>
<td>4 +</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>5</td>
<td>Data submission</td>
<td>4 +</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>5</td>
<td>Delete reports</td>
<td>2 +</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>5</td>
<td>Accept reports</td>
<td>4 +</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>2</td>
<td>Integration of iGDACS</td>
<td>5 +</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>2</td>
<td>Interfaces for data access</td>
<td>8 +</td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>1</td>
<td>Installation and deployment guide</td>
<td>6 +</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>1</td>
<td>User handbook</td>
<td>8 +</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>1</td>
<td>Test</td>
<td>36 +</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Work packages with efforts in man-days and priorities
Figure 24 summarizes the planned project phase and intensity cycle according to the RUP iterative development process. A detailed project plan consisting of individual timetables for detailed implementation tasks will be prepared when a final agreement on the GDACSmobile functions is reached.

Figure 24: project phases and discipline intensity
### Authors

<table>
<thead>
<tr>
<th>Name</th>
<th>E-Mail address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carsten Bubbich</td>
<td><a href="mailto:carstenbubbich@yahoo.de">carstenbubbich@yahoo.de</a></td>
</tr>
<tr>
<td>Friedrich Chasin</td>
<td><a href="mailto:f_chas01@uni-muenster.de">f_chas01@uni-muenster.de</a></td>
</tr>
<tr>
<td>Sven Kronimus</td>
<td><a href="mailto:sven.kronimus@uni-muenster.de">sven.kronimus@uni-muenster.de</a></td>
</tr>
<tr>
<td>Stefan Laube</td>
<td><a href="mailto:Stefan.Laube@uni-muenster.de">Stefan.Laube@uni-muenster.de</a></td>
</tr>
<tr>
<td>Hendrik Petersen</td>
<td><a href="mailto:petersen.hendrik@uni-muenster.de">petersen.hendrik@uni-muenster.de</a></td>
</tr>
<tr>
<td>Philipp Saalmann</td>
<td><a href="mailto:p.saalmann@uni-muenster.de">p.saalmann@uni-muenster.de</a></td>
</tr>
<tr>
<td>Martin Vanauer</td>
<td><a href="mailto:martin.vanauer@uni-muenster.de">martin.vanauer@uni-muenster.de</a></td>
</tr>
</tbody>
</table>
Appendix A: Twitter Processing Details

Conceptual Background
The twitter processing solution within GDACSmobile is explained in the context of a simplified model which is capable of describing any process aimed to make text-based social media content useful in disaster management. The model consists of exactly 6 stages.

1. Informing
2. Posting
3. Fetching
4. Analyzing
5. Saving
6. Displaying

In the stage 1) users of social media can be informed about the way they can use social media to share their knowledge/observations. In stage 2) users create a message. During the stage 3) social media data is fetched through the use of a corresponding API. Different sources allow for different search queries. The goal is to make a first separation between relevant and irrelevant information. Stage 4) is the one with the highest variety of approaches and techniques which can be applied. There are two main tasks within this step. The first is to filter the data. The second is to categorize data, in order to mark it with tags or to enrich it. After data is analyzed, it is saved in stage 5) using an appropriate format so it can be finally displayed in stage 6)

Algorithm Overview
The GDACSmobile short messages processing focuses on analyzing Twitter data. It relies on extensive use of the informing stage (1) and crowdsourcing integration during the analysis stage (4). During the informing stage, the format of the messages (which is basically just a combination of hashtags) is broadcasted to the affected population by official sources like for example the JRC itself(using text messages, radio, Twitter, facebook etc. ). This format is then used by the affected population to create and share messages, thus allowing machines to categorize the messages in the analysis stage without automatic learning mechanisms. For messages with partly or completely missing hashtags from the defined format, categorization can still be made manually by involving different types of users to classify the messages manually. This manual classification will be performed directly on the GDACSmobile website as outlined in the functionalities chapter.

Benefits
The main benefits of the algorithm are its universality and flexibility regarding different disaster types. Although message format broadcasting is a promising approach, it is crucial to mention that the described Twitter message processing does not necessarily depend on broadcasting. By providing a service for manual crowd categorization, the GDACSmobile application will always offer a solid Twitter integration even without any automating categorization efforts. Last but not least, the algorithm will store Twitter information in the same format as data coming in from the needs assessment functionality of GDACSmobile.

Algorithm Details
Input / Output
The algorithm gets two different inputs.

1) Mapping between broadcasted hashtags and internal categories within the GDACSmobile application. For example #infra can be associated with the category ‘infrastructure’.
2) Fetching criteria for Twitter API search requests. (different search criteria such as hashtags but also time and location criteria can be used)

The output of the algorithm is a set of data records, which will be stored into a database containing categorized pieces of information from Twitter.

Processing
The algorithm’s way of functioning is represented in the Figure 25.

![Figure 25: Algorithm processing](image)

The starting point of the algorithm is a set of Twitter messages shared on the Twitter platform. A part of the messages is unrelated to the search criteria provided for the algorithm as input. Another part corresponds to the search criteria. Furthermore, there is a subset of messages which were crafted according to the broadcasted format from the first stage.

After fetching, tweets which do not satisfy the search criteria are filtered out. Fetched tweets are the subject of the succeeding analysis. First, fetched messages can be pruned according to some additional criteria (for example due to too short length of the message, certain location or due to the occurrence of the author in a “black list”) (see 4.0 in Figure 25). Afterwards, the content of the message is parsed and scanned for occurrence of broadcasted hashtags (which are provided for the algorithm as a mapping, see Input/Output). Based on found hashtags, categorization is made for each message. This results in two distinct sets of messages. The first one consists of those messages which provided enough broadcasted hashtags in order to be categorized in a satisfactory way. The second one represents messages which require manual categorization.

1A possible search query using the Twitter API can look like this:
http://search.twitter.com/search.json?q=#californiafire&result_type=mixed
Manual categorization is made directly using GDACSmobile form (at the server side). Messages are displayed to (authorized) users at the server side who can manually add missing category assignment to those messages (4.5 in Figure 25).

Both, automatically categorized messages and those edited by the crowd are saved into the database in a format allowing visual representation of the message in the GDACSmobile application. All saved messages are subject of verification by special user groups such as JRC moderators (5.5 in Figure 25).

**Structured View on the Algorithm**

Figure 26 illustrates the structogram of the tweet processing.

![Figure 26: Twitter processing structogram](image)

**Possible Scenario**

An earthquake occurred in location X. There are two hashtags which represents the majority of all related twitter messages: #X_earthquake and #Xhelp. Both hashtags are set by JRC moderators as a search criteria for the algorithm (JRC defines relevant search criteria directly over a restricted area of GDACSmobile web form). All messages satisfying the two hashtags are fetched using the Twitter API. Requests to the server happen continuously, e.g. every 5 Minutes. In the first 5 Minutes 23 messages are fetched. 7 out of these 23 messages are filtered out due to formal criteria (location, length of the message etc., the list of criteria is a subject of current design). The remaining 16 messages are displayed on the GDACSmobile website to the registered crowd analysts. Users are intuitively categorizing the messages with the help of a provided GUI on the website. Results are stored in the database and are ready to be verified by JRC moderators and then can be retrieved by the GDACSmobile application or any other applications which integrate GDACSmobile data.

After 2 hours, there is information about water supply problems. GDACS decides to broadcast a Twitter message format for reporting water supply issues.

```
@GDACSmobile: X_earthquake, #Xhelp. Please use following hashtags to report water supply needs. #ws, #city [city], #street [street]
```

The message is submitted via the GDACSmobile website together with a mapping between #ws and water supply category. Now a message containing this particular hashtag will be automatically
categorized as water supply issue report and can be directly stored in the database avoiding the crowdsourcing step. Every record in the database is still the subject for verification by JRC moderators such that only verified tweets are displayed in the application.