Development of an Application for Indoor Navigation and Deviation Detection

Supervisor: Prof. Sara Comai
Co-Supervisors: Emanuele De Bernardi

By:
Reyhane Monajemi (796824)

Academic Year 2015-2016
Acknowledgment

I cannot express enough thanks to my thesis supervisor Prof. Sara Comai, for her continued supports and encouragements. My thesis could not have been successfully conducted without her very valuable comments on this thesis.

Also I must express my very profound gratitude to my parents and to my loving family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Thank you.
Abstract

Statistical analysis show that the ratio of elderly people to total population is increasing at an unprecedented rate all over the world. Insofar as the global population of people with 60+ years old will be growing by three times and will reach to almost 2 billion in 2050. Aging, however, comes with a series of physical and mental disabilities which may hinder daily activities. But theses disabilities should not stop the older adults to perform their routine tasks because majority of them want to keep their independency in performing those tasks. New developments in assistive technologies make a great contribution in improving the elderly people’s quality of life. Different elderly care applications have been designed to fulfil this purpose and among those, the most useful ones are for indoor navigation assistant.

This thesis focuses on implementation and development of a user interface for an application which provides an indoor navigation assistant specifically for elderly users of ALMA project by considering users requirements. The aim of ALMA is to combine a set of advanced hardware and software technologies into an integrated, non-invasive and modular system in order to offer assistance to people affected by different types of impairments. In the ALMA context end-users can be either patients or healthcare personnel.

Within this project, we implement and develop the Alma user-friendly application by considering users’ needs and limitations. Our solution provides different services such as:

- Indoor navigation assistant using which users can get direction to a specific destination and get graphical and vocal guide during the path
- Agenda plan with regular reminder notification about each plan
- Messaging and calling services using which users can contact directly to the supervisor in case they need assistance by voice

By researching and analyzing the Alma system, we introduce, design and implement four new innovative deviation algorithms for the navigation system, practical and functional to the system, as follows:

- Delay Deviation
- Temporal Deviation
- Orientation Deviation
- State Deviation

These algorithms diagnose and intercept possible deviations during navigation. They help the user get to the desired destination on time and detect if they have been deviated from the path and informs them using different haptic, audio and text notifications.
Table of Contents

1. Introduction .............................................................................................................................. 9
   1.1. Objectives of the thesis ...................................................................................................... 10
   1.2. Structure of the thesis ...................................................................................................... 11

2. Overview of the ALMA Project .............................................................................................. 12
   2.1. The ALMA Project .......................................................................................................... 12
   2.2. Personal Navigation Assistant: a System Overview ....................................................... 13
   2.3. Alma IndoorGML .............................................................................................................. 14
   2.4. Recent developments done on Alma project ................................................................. 18

3. Requirement Analysis ............................................................................................................. 19
   3.1. General Requirements of the PNA ................................................................................ 19
       3.1.1. Device characteristics ................................................................................................. 19
       3.1.2. Safety and Reliability ............................................................................................... 21
       3.1.3. System Customization ............................................................................................... 21
       3.1.4. Usability .................................................................................................................... 21
   3.2. Functionality requirements of the PNA .......................................................................... 22
       3.2.1. Users of the Personal Navigation Assistant ............................................................. 22
       3.2.2. Main Functionalities .................................................................................................. 23
   3.3. Navigation Support ......................................................................................................... 24
       3.3.1. Visualization of a destination .................................................................................... 24
       3.3.2. Selection of a destination .......................................................................................... 26
       3.3.3. Send the request for a route to reach a selected destination .................................... 26
       3.3.4. Visualization of information about a route ................................................................. 27
       3.3.5. Get directions to reach the destination ..................................................................... 28
       3.3.6. Detect deviations from the expected path ................................................................. 29
   3.4. Other Functionalities ....................................................................................................... 30
       3.4.1. Call request ................................................................................................................ 30
       3.4.2. Message request ...................................................................................................... 30
       3.4.3. Visualization and selection of the available functionalities ..................................... 31
       3.4.4. Interface configuration ............................................................................................... 31
       3.4.5. Agenda management ................................................................................................ 31
       3.4.6. Notifications reception from the ALMA system ..................................................... 32
       3.4.7. Information and Feedback about the Path ............................................................... 32
## Table of Figures

- **Figure 1** Population aged 60 or over: world and development regions, 1950-2050
- **Figure 2** Functional modules of the Alma system and their relationships
- **Figure 3** L0 – TOPOGRAPHIC – Geometry
- **Figure 4** L1 – TAGS – Semantic
- **Figure 5** L2 – SENSOR – Localization
- **Figure 6** L3 – TOPOGRAPHIC – Navigation
- **Figure 7** L4 – SENSOR – Camera
- **Figure 8** W – TOPOGRAPHIC – Navigation
- **Figure 9** Market share worldwide smartphone shipments by operating system from 2014 to 2020
- **Figure 10** Pie chart of Android OS versions distribution
- **Figure 11** User's hierarchy
- **Figure 12** Examples of output modality for the destinations with visual icons and text
- **Figure 13** Example of visualization of information about the route
- **Figure 14** Example of UI for the navigation
- **Figure 15** Sample of the former interface pages
- **Figure 16** Prototype presentation
- **Figure 17** Login page
- **Figure 18** User Nurse Login page
- **Figure 19** Choose Remember checkbox
- **Figure 20** Registration page
- **Figure 21** Notification about filling out all the fields of registration page
- **Figure 22** Main page
- **Figure 23** Emergency call page
- **Figure 24** Message page
- **Figure 25** Agenda page
- **Figure 26** Agenda detail page
- **Figure 27** Navigation page
- **Figure 28** Destination detail page
- **Figure 29** Navigating page
- **Figure 30** Feedback page
- **Figure 31** Nurse page
- **Figure 32** Setting page
- **Figure 33** Change password page
- **Figure 34** Edid profile page
- **Figure 35** Active/Inactive call and message
- **Figure 36** Delay deviation algorithm
- **Figure 37** Temporal deviation algorithm
- **Figure 38** Orientation deviation algorithm
- **Figure 39** State deviation algorithm
Figure 40 Delay-Deviation notification ..............................................................70
Figure 41 Tempora-Deviation notification ..........................................................71
Figure 42 Orientation-Deviation notification .......................................................71
Figure 43 State-Deviation notification .................................................................72
1. Introduction

According to the researches [1], the number of elderly and their percentage to total population is sharply rising all over the world. In 1950, there were 205 million persons aged 60 or over throughout the world (Figure 1). Fifty years later, in 2000 the number has increased about three times to 606 million. Over the first half of the current century, the global population of people with 60+ years old is projected to expand by more than three times to reach nearly 2 billion in 2050.

![Figure 1](image.png)

**Figure 1** Population aged 60 or over: world and development regions, 1950-2050

On the other hand, aging will cause physical and mental disorders which cause inchmeal reduction of the performance in activities thus people might face some difficulties to perform their daily duties or personal tasks. But beside all of these barriers, a majority of older adults hope to preserve their independency, especially in their routine daily activities.

New developments in assistive technologies are likely to make an important contribution to improve life quality for elderly people in all aspects of their lives. In that context, elderly
care applications perfectly match this idea and one of the most common and useful applications is an indoor navigation assistant.

1.1. Objectives of the thesis

The goal of this thesis is to design and develop a user interface of a Personal Navigation Assistant (PNA): the PNA is one of the modules that will be provided by ALMA, an international project within the Ambient Assisted Living (AAL) Joint Programme, started on April 2, 2013 and involving the following partners: Scuola Universitaria Professionale della Svizzera Italiana (Switzerland), Politecnico di Milano (Italy), Infosolution SpA (Italy), VCA Technology Ltd. (UK), Istituti Sociali di Chiasso (Switzerland) (ICS in the sequel), Clinica Hildebrand (Switzerland) (HILDE in the sequel), University of Wuerzburg (Germany), Degonda SA (Switzerland).

The aim of ALMA is to combine a set of advanced hardware and software technologies into an integrated, non-invasive and modular system in order to offer assistance to people affected by different types of impairments. In the ALMA context end-users can be either patients or a healthcare structure.

This thesis provides an indoor navigation assistant for elderly users of ALMA project. The focus of design and implementation is to represent the Alma user-friendly interface application by considering users different needs and limitations, which provides messaging, calling, agenda and navigation services to the users.

In this thesis, we improve and extend the previously designed interface of the ALMA application, according to the requirements of the system provided by our end partners by introducing a mockup interface. Then implement the interface with connection to the server, so that by interaction with the server, we can get/post required data from/on the server database. This interface platform provides agenda plan with daily reminders to the users. It also provides indoor navigation assistant which users can be directed to a specific destination and get vocal guide during the path. We also introduce innovative deviation algorithms categorized into four groups of “Delay Deviation”, “Temporal Deviation”, “Orientation Deviation and “State Deviation” for the navigation system, to diagnose and intercept possible deviations during navigation.

The main contributions of the thesis are:
• Designing a user-friendly interface application which provides different services such as: Agenda, Messaging, Calling and Navigation

• Sending two notifications about the Agenda plan to users in determined interval time before its due time

• Providing navigation assistant system together with vocal guide during the path

• Designing and implementing some deviation algorithms to detect possible deviations during the path and react against them

1.2. Structure of the thesis
In Chapter two, we overview Alma system and Personal Navigation Assistant (PNA). In Chapter three, we analyze and investigate the general requirements and functional requirements of the Alma system. In Chapter four, we describe the interface design and the interface implementation. In Chapter five, we explain about our new innovative deviation algorithms. In Chapter six, the implementation parts of the project are explained. In chapter seven, the summary of the work and future work are discussed.
2. Overview of the ALMA Project

Among the difficulties that old age and disability force upon people, those related to mobility have an especially high impact on quality of life and psychological wellbeing. The Alma project [2] is an international project aim to combine a set of advanced hardware and software technologies into an integrated, non-invasive and modular system in order to offer assistance to people affected by different types of impairments, proposes a personal navigation assistant supporting user friendly interface to all the functionalities of the system, tailored to the special user-defined requirements and physical limitations (e.g., vocal and tactile interfaces, ad-hoc devices). Alma will bring to the mobility of a wide range of primary end-users a real advancement that will be measured in terms of the most appropriate metric: their own feeling of freedom and increased empowerment. At the same time, secondary end-users, e.g., residences and hospitals will leverage on the information collected by the system on the movements of their guests to monitor their aging, to design personalized support services or rehabilitation paths.

2.1. The ALMA Project

The Alma project tackles the issue of not being able to move autonomously or efficiently by combining a set of advanced hardware and software technologies into an integrated and modular system composed by:

- An indoor localization system based on a network of low-cost/low-power RF emitters, to provide room level localization of people and objects
- An ad-hoc, autonomic HW/SW system based on networked smart cameras providing accurate indoor and outdoor localization and environment monitoring;
- An intelligent system for the online planning of users’ paths according to their specific needs, matching these with the actual state of the environment and the available resources;
- A personal mobility kit for electric powered wheelchairs allowing them to perform automatic or assisted navigation and, additionally, to interact with the surrounding environment;
- A personal navigation assistant supporting user-friendly interface to all the functionalities of the system, tailored to the specific user defined requirements and physical limitations (e.g., vocal and tactile interfaces, ad-hoc devices).
Alma users will be supported in their mobility to acquire knowledge about interesting locations (e.g., services, people, facilities, etc.), to select and follow an efficient and safe path to such destinations considering their needs and/or limitations or the status of the environment, to present the resources provided by intelligent environments to the users, so they can effectively access them with familiar instruments without feeling disoriented or overwhelmed by technology. Alma aims at bringing to the mobility of a wide range of primary end-users a real advancement that will be measured in terms of the most appropriate metric: their own feeling of freedom and increased empowerment. At the same time, secondary end-users, e.g., residences and hospitals will leverage on the information collected by the system on the movements of their guests to monitor their aging, to design personalized support services or rehabilitation paths.

### 2.2. Personal Navigation Assistant: a System Overview

The **Personal Navigation Assistant** (PNA) consists of a user-friendly interface to all the functionalities of the ALMA system, tailored to the limitations (if any) of the specific user; the PNA is the main interface towards the ALMA system provided to the primary end user. It is a software application for standard devices (e.g., smartphones or tablets) which:

- assembles requests for mobility plans and sends them to the Integration and Planning Module (IPM);
- receives back a global plan/schedule, and guides the user in putting it into effect using suitable suggestions;
- monitors the progress of the user through the plan, and, depending on the user and on the local circumstances (e.g., presence of unexpected obstacles) supports the user in modifying the plan, and when necessary, also automatically requesting a plan;

The other modules of the ALMA system with which the PNA needs to interface and collaborate are depicted in Figure 2 and include:

- An **Integration and Planning Module** (IPM) of users' paths according their specific needs, to available resources and to the state of the environment, the planning module, which is the core of the ALMA system, is the place where all available information and requests converge are processed to constantly adapt the behavior of the system both to user necessity and to the actual state of the environment.
- The **Personal Mobility Kit** (PMK) which turns an existing commercial wheelchair into a device capable to provide assistance to the user through autonomous and semi-autonomous functions, when required by the user. Such functions are configurable, and range from assisting the driver (e.g., to cancel the effect of tremors
from the commands issued to the wheelchair’s motors) to enhancing safety (e.g., by
detecting and preventing possible collisions) to autonomously navigating towards a
destination.

Figure 2 Functional modules of the Alma system and their relationships

Further information can be obtained – possibly indirectly through the planning module -
from the following subsystems:

- The Localization Module (LM) dedicated to detecting the position of specific items
  (e.g., a user) within the environment exploiting wireless technologies and signal
  processing. This is an important component for all the functionalities of the ALMA
  system, because knowing the physical location of interesting people, places and
  objects is essential to path planning and navigation.
- A system of networked smart cameras and servers providing indoor and outdoor
  video based monitoring (VM) of the environment (the camera-based monitoring
  module provides the planning module with continuously updated information about
  the environment(s) where ALMA users move.

2.3. Alma IndoorGML
IndoorGML is a candidate OGC standard for an open data model and XML schema for
indoor spatial information. It aims to provide a common framework of representation and
exchange of indoor spatial information. It is defined as an application schema of OGC
Geographic Markup Language (GML) 3.2.1 [3].
IndoorGML is used in ALMA project for storing the map information. ALMA IndoorGML
contains six different layers represented in the following:
**L0 – TOPOGRAPHIC – Geometry Layer**: Represent geometrical information of indoor spaces (Figure 3)

![Figure 3 L0 – TOPOGRAPHIC – Geometry](image)

**L1 – TAGS – Semantic Layer**: Represent semantic information of spaces using a unique label (Figure 4)

![Figure 4 L1 – TAGS – Semantic](image)
**L2 – SENSOR – Localization:** Represent the localization sensors and their range of coverage (Figure 5)

![Figure 5 L2 – SENSOR – Localization](image)

**L3 – TOPOGRAPHIC – Navigation:** Represent path planning information in details and location of each cell (Figure 6)

![Figure 6 L3 – TOPOGRAPHIC – Navigation](image)
**L4 – SENSOR – Camera**: Represent the localization of camera sensors (Figure 7)

![Figure 7 L4 – SENSOR – Camera](image)

**W – TOPOGRAPHIC – Navigation**: Represents path planning information with detail about the states and partitioning states of corridors, rooms and doors (Figure 8)

![Figure 8 W – TOPOGRAPHIC – Navigation](image)
2.4. Recent developments done on Alma project

The difficulties existed for elderly people in using the smartphone and the analysis of innovative technologies, to answer the demands of elderly have already been investigated in [4]. She has also designed a first basic of prototype for the user interface of application of Alma project according to the requirements of the ALMA system.

This thesis mainly focuses on improvement of already prepared interface with respect to our end partners’ requirements and suggestions plus the implementation of the interface application providing the navigating together with monitoring the progress of user through the path considering the feasible circumstances and also providing audio suggestions through the path, which also can detect and intercept possible deviations from the path.
3. Requirement Analysis

3.1. General Requirements of the PNA

3.1.1. Device characteristics

The PNA will be developed for handheld devices like smartphones or tablets.

As operative system to support the PNA we will use Android. Indeed, Android is currently the most diffuse mobile OS. Figure 9 shows the market share held by smartphone operating systems of smartphone unit shipments worldwide from 2014 to 2020 [5]. In current year 2016, the Android operating system is forecast to hold 83.76 percent share of smartphone shipments.

![Figure 9](image)

**Figure 9** Market share worldwide smartphone shipments by operating system from 2014 to 2020

Given the length of the project the latest versions of Android will be considered. According to the current trends [6], as shown in Figure 10, the Marshmallow version will be considered and compatibility with other device platforms will be guaranteed at least starting from Android version 3.0.
The minimum API version of the Android platform on which the application is guaranteed to run is 11. The current available level is 23, given the length of the project the current development will start from this level.

Since different sensors will be used to develop the required functionalities, the PNA is designed in such a way that the latest sensors (e.g. the barometer) can be ignored: however, in such cases, less precise results might be obtained for some specific functions.

Considering the available phone capabilities, the following sensors and phone capabilities are required:

- Gyroscope: required
- Digital compass: required
- Accelerometer: required
- Barometer: useful for some functionalities, but optional – possibly less precise results might be obtained
- Wi-Fi: required for interactions with the ALMA system
- Bluetooth: required for interactions with the PMK
- GPS: required for outdoor navigation
• Microphone: required for audio input modality

3.1.2. Safety and Reliability
As the PNA will be used by users with special needs, a high level of accuracy is necessary, to reduce errors as much as possible.

Accuracy and reliability are needed for various functionalities:

• The user holding the PNA shall be correctly localized, at least at the room-level
• The user holding the PNA shall be correctly guided to the desired destination
• The PNA with the help of the IPM and the LM modules shall be able to detect if the user is not following the given route.

3.1.3. System Customization
Users (or their supervisors) shall be able to set some preferences in the application.

In particular:

• The set of offered functionalities shall be customizable depending on the users’ needs or supervisor’s preferences: for example, it shall be possible to active/inactive functionalities like, for example, calling or messaging.
• The PNA shall take into account the type of user and provide him/her a personalized interface: for example, a wheelchair user will see different information (e.g., only accessible destinations/areas) compared to a walking user.
• The information shown to the user shall take into account also the user’s physical and perceptual capabilities. For example, the size and the color of the font should be personalized.
• Different interaction modality shall be supported:
  o Input modalities shall include: touch-based
  o Output modalities shall include: text, visual, and audio
• The application shall be multilingual: the language shall be selected by the user; extendibility to other languages shall be supported.

3.1.4. Usability
Elderly people have decreasing learning capabilities and may show fatigue in learning the usage of new technologies.
The PNA shall realize a very simple and easy to use interface, which will display only the needed information in the best modality for the user.

Also the need for training should be limited.

The complexity must therefore stay "behind" the device.

The following aspects should be guaranteed:

- Ease of learning
- Adherence of the interaction objects (menus, buttons, etc.) to well-known standards, both for mobile interaction (in terms of typical mobile widgets) and also for the target users
- Coherent use of the interaction across the whole application
- Availability of mechanisms for orienting and assisting the user when using the application
- Considering different devices which have different screen sizes and different resolutions

3.2. Functionality requirements of the PNA

3.2.1. Users of the Personal Navigation Assistant

The following groups of users will use the application:

- Nurses: who have access to the administration page in which they can:
  - Edit the information of each user
  - Manage the settings of the application
  - Remove an instance user
  - Disable/Enable the possibility of messaging and calling for each user

- Walking users: the ordinary target users of the PNA with access to the main page and corresponding ones

- Wheelchair users: the ordinary target users of the PNA with access to the main page and corresponding ones
3.2.2. Main Functionalities

Depending on the user, different functional requirements are needed. Here they are briefly summarized and explained in more details in the next subsections.

**NAVIGATION SUPPORT**

**Target users: ALL**

<table>
<thead>
<tr>
<th>FUNCTIONALITY</th>
<th>SHORT DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visualization of the possible destinations</td>
<td>The destinations specified for the current user are displayed on the device.</td>
</tr>
<tr>
<td>Selection of a destination</td>
<td>Given a set of destinations from which to choose, the user selects a destination.</td>
</tr>
<tr>
<td>Send the request of a route to reach a selected destination</td>
<td>Given a selected destination a plan to reach it can be requested.</td>
</tr>
<tr>
<td>Visualization of information about</td>
<td>Given a plan to reach a selected destination, the PNA provides him some information about destination such as title, estimated time, description and possibly to use audio to listen to the description of destination and also possibly to allow him/her to explore the route.</td>
</tr>
</tbody>
</table>
Get directions to reach the destination | Given a path, directions are provided to the user, with different output modalities (Graphic and voice)
Detect deviations from the expected path | When following a path, any change must be managed by the ALMA system. Interaction with the user and with the ALMA system is required.
Record information and feedback about the path | When a destination has been reached the user/supervisor can provide feedback about the navigation system.

OTHER FUNCTIONALITIES

Target users: ALL

Check agenda / next events | The PNA provides the time and description of the next event/activity. The user can also check the description of the path to reach it and possibly start following a path.
Help request | Through the PNA the user can request help to the system. The user can make an emergency call or send message to the nurse.
Visualize and select available functionalities | All the available functionalities should be accessed through a menu.
Interface configuration | The interface of the PNA should take into account the characteristics of the user and the modalities shall be personalized.
Receive notifications from the ALMA system | The device can receive notifications from the system about: have an agenda

3.3. Navigation Support

3.3.1. Visualization of a destination
The user shall check the list of possible destinations.

Destinations shall include:
- Specific rooms: examples of possible destinations may be the dining room, the bedroom, the bathroom, etc.
- Other places/facilities: park, café, restaurant, animation point, chapel, hairdresser, day center animation, therapeutic day center, gym, etc.

The PNA considers destinations only at the semantic level (only the name of the destination is provided to the user).

The available destinations shall be visualized by means of icons and text.

![Figure 12 Examples of output modality for the destinations with visual icons and text](image)

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Visualization of a destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions</td>
<td>The ID of the user is known.</td>
</tr>
<tr>
<td>Postconditions</td>
<td>The device displays the list of available destinations</td>
</tr>
<tr>
<td>Description / Steps</td>
<td></td>
</tr>
<tr>
<td>The device sends to the ALMA system the ID of the user</td>
<td></td>
</tr>
<tr>
<td>The device receives from the ALMA system:</td>
<td></td>
</tr>
<tr>
<td>- the list of destinations</td>
<td></td>
</tr>
<tr>
<td>- the list of agenda</td>
<td></td>
</tr>
<tr>
<td>The PNA checks if according to the agenda of the day there is an event/activity scheduled in the next half-hour that should be suggested to the user</td>
<td></td>
</tr>
</tbody>
</table>
3.3.2. Selection of a destination
The user may be able to ask for a route (e.g., to go the gym, to the dining room, back to the sleeping room), from his/her current position.

The selection of the destination is supported by the PNA.

Navigating list of destinations and selecting a destination is specified using touch input.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Selection of a destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions</td>
<td>A list of destinations is displayed on the device.</td>
</tr>
<tr>
<td>Postconditions</td>
<td>A specific destination is selected.</td>
</tr>
<tr>
<td>Description / Steps</td>
<td>Given the list of available destinations the user can select one of them.</td>
</tr>
<tr>
<td></td>
<td>The selected destination is shown to the user.</td>
</tr>
</tbody>
</table>

3.3.3. Send the request for a route to reach a selected destination
The PNA can send to the IPM the request for a given destination.

The request may be sent upon explicit user’s request (for example, when the user checks his/her agenda and for a selected activity the route to the destination is requested)

The route starts from the current position of the user.

The current position is computed by the ALMA system and provided to the PNA through the IPM module.

All the data needed to compute the route to the destination for the given user are stored in the ALMA system. To get the route, the PNA sends to the IPM the user’s position (as start point) and the selected destination position (as end point).

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Route request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions</td>
<td>A specific destination is selected.</td>
</tr>
<tr>
<td>Postconditions</td>
<td>The device receives the route from the current position to the selected destination.</td>
</tr>
<tr>
<td>Description / Steps</td>
<td>The PNA sends the selected destination position and user position to the ALMA system. The ALMA system returns the route from the current position to the selected destination.</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>

### 3.3.4. Visualization of information about a route

Once the user receives a route to reach a destination, some information such as (Estimated) time needed to reach the destination may be displayed.

This step may be possibly skipped and the directions to the destination directly provided.

![Figure 13 Example of visualization of information about the route](image)

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Route visualization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions</td>
<td>A route is available</td>
</tr>
<tr>
<td>Postconditions</td>
<td>Information about the route is visualized</td>
</tr>
<tr>
<td>Description / Steps</td>
<td>The information about the route provided by the ALMA system is visualized (e.g. estimated time)</td>
</tr>
</tbody>
</table>
3.3.5. Get directions to reach the destination

Once a route is available and it is started, the PNA provides directions to the user.

In order to provide the correct directions, the PNA needs to

- Understand the current orientation/heading of the user
- Provide directions on the basis of the current position of the user

Directions will be provided by exploiting the localization module.

The output of this functionality will be provided through a simple interface. An example of a very simple interface is shown in Figure 14, where only the direction to follow is shown.

![Figure 14 Example of UI for the navigation](image)

Output modalities: beside the graphical output also audio output shall be provided, with predefined commands like (“go to the lift”, “go through the corridor”, “go to the door”, etc.).

| Functionality       | Directions to the destination |
**Preconditions**
A route is available and the user has started following a path.
Hypothesis: the user correctly follows the path until the destination

**Postconditions**
Directions to the destination are provided

**Description / Steps**
Starting from the current position
1. Directions to the next waypoints are provided
2. The PNA continuously check the current position of the user (periodically requested to the ALMA system) wrt to path to follow
   a. If the destination is reached the PNA informs the user – the process stops.
   b. If a waypoint is reached the process continues from step 1 considering the next waypoint.

**Exceptions**
Deviations from the expected path (see description below)
Notifications from the ALMA system (see the corresponding functionality)
The request may be deleted or suspended (to be reactivated later).

**Other related functionalities**
During the navigation of a path:
- The PNA periodically asks the server an up-to-date plan

### 3.3.6. Detect deviations from the expected path

When following a path, any possible deviation from the expected behavior must be identified and managed.

The PNA periodically asks the current position of the user and checks if the user is on the path / detects if a change of direction wrt to the expected path occurs.

If after a given time/travelled distance the user is not on his/her path the user is notified.

Possible behaviors include:
- The user is notified;
- The user is asked to continue/suspend/cancel the route;
- The user may ask help to the ALMA system so that the staff can be informed;
3.4. Other Functionalities

3.4.1. Call request

While using the application the user may need to make a call to his/her supervisor for some help/request.

When the system does not work/has problems or the user does not understand where to go the user can (explicitly) ask for a call.

In particular, a call button shall be provided on the interface to make a call to supervisors when needed.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Call request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions</td>
<td>The call button is displayed on the PNA and the user clicks on it</td>
</tr>
<tr>
<td>Postconditions</td>
<td>A call is sent to the supervisor</td>
</tr>
<tr>
<td>Description / Steps</td>
<td>When the user touches call button, the supervisor receives the phone call and proceeds accordingly</td>
</tr>
</tbody>
</table>

3.4.2. Message request

While using the application the user may need to send a message to his/her supervisor to ask for some help/request.

The message button shall be provided on the interface to send a message to supervisors when needed.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Message request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions</td>
<td>The Message button is displayed on the PNA and the user clicks on it</td>
</tr>
<tr>
<td>Postconditions</td>
<td>A Message is sent to the supervisor</td>
</tr>
<tr>
<td>Description / Steps</td>
<td>When the user touches message button, he/she can write the message buddy and send it. The message is directly send to the supervisor and he/she proceeds accordingly</td>
</tr>
</tbody>
</table>
3.4.3. Visualization and selection of the available functionalities

All the functionalities available to the user (e.g., reach a destination, check agenda, call, etc.) shall be accessed through a menu.

It shall be possible to configure the application to select the functionalities that are made available to the user, i.e., the content of the menu shall be configurable (e.g. nurses with admin access can choose to active/inactive some functionalities such as call, message for some users).

3.4.4. Interface configuration

It shall be possible to configure the interface of the PNA at least with respect to:

- Available functionalities: it shall be possible to configure the activation/inactivation of the different functionalities offered to the user (e.g. active/inactive call or message)
- User’s impairment: it shall be possible to configure the interface for different users with different impairments (e.g. big font size for vision impairment users)

3.4.5. Agenda management

The user can check the agenda of the day. In particular, the next event/activity in the agenda can be checked as well as the whole list of activities of the day.

Once, an activity is selected, the PNA provides details about the time and description of the activity.

The user can also request the path to reach it (from the current position).

For the next activity/event in the agenda, the path can be activated to reach the destination.

Different input/output modalities shall be supported, in a similar way to the functionalities described above.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Check agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions</td>
<td>The PNA receives the agenda of the day of the user from the PNA</td>
</tr>
<tr>
<td>Postconditions</td>
<td>The whole list of events or a particular event can be checked</td>
</tr>
</tbody>
</table>
Given the agenda of the day, the PNA displays the whole list of events. The user can select a specific event/activity and check details like title, time and description. It shall be possible to see in advance also the route to reach the destination.

3.4.6. Notifications reception from the ALMA system

The device shall be able to receive notifications from the system, for example, to:

- Notify the user about the agenda of the day: So that user can get ready on time for that plan

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Notifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions</td>
<td>Always available</td>
</tr>
<tr>
<td></td>
<td>The ALMA system sends a notification to the PNA</td>
</tr>
<tr>
<td>Postconditions</td>
<td>The PNA receives the notification</td>
</tr>
<tr>
<td>Description / Steps</td>
<td>The ALMA systems sends a notification to the PNA.</td>
</tr>
</tbody>
</table>

3.4.7. Information and Feedback about the Path

During the experimentation at the end of navigation, the user will be asked to provide some feedback about the path, the received directions, etc.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Record information and feedback about the path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions</td>
<td>A destination has been reached</td>
</tr>
<tr>
<td>Postconditions</td>
<td>Data about the path are recorded</td>
</tr>
<tr>
<td>Description / Steps</td>
<td>The user (possibly with the help of the staff) is required to rate the service. Collected data are sent to the ALMA system.</td>
</tr>
</tbody>
</table>

3.4.8. Error cases

All the above functionalities assume that the system works correctly. However, in practice different cases of errors may occur.
The following error cases shall be considered by the PNA application:

- **Errors caused by the user:**
  - In case the user try to send an empty message to the supervisor
  - In case of leaving an empty field in registration page
  - In case a user try to login to the PNA as nurse using wrong password

- **Errors related to the network:**
  - In case of possible network problems errors shall be notified to the user

### 3.5. PNA Use cases

<table>
<thead>
<tr>
<th>Use case ID</th>
<th>PNA-walking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary actor</td>
<td>Walking user</td>
</tr>
<tr>
<td>Pre-condition</td>
<td>The system is correctly working and the PNA is active</td>
</tr>
<tr>
<td>Post-condition</td>
<td>The user reaches the requested destination</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flow of events</th>
<th>User</th>
<th>PNA</th>
<th>ALMA system</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selection of the agenda functionality</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visualization of the agenda o</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select an activity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show details (title, time, description) of the next activity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ask to go to the destination of the selected activity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request path to the destination</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Return path</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show directions until the next waypoint</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The PNA interacts with the ALMA system to check the position of the user along the path</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Follow directions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The PNA interacts with the ALMA system to check the position of the user and to detect the reaching of the waypoint</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Follow directions until destination</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inform the user that the destination has been reached</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collect feedbacks from the user</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insert feedback</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Send feedback to the ALMA system</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminate the activity and visualize the main application screen.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.8. PNA Scenarios

Wheelchair User

Name & Family: Antonio Salvatore

Age: 85

Disability: Walking disability – memory difficulty

I have to go to the physiotherapy room every other day and also need to get pills 2 times a day, but I have some memory problems and sometimes it is hard for me to remember the way to the physiotherapy room or when to get my pills!

Use of Technology

Smart Phone: 80%     Tablet: 70%         PC: 30%

Antonio is an active user of Internet. He has Facebook account to get in touch with his friends and families and usually uses his tablet surfing the web. Also since he has Memory difficulty, always need to have his phone is his pocket in case of needing help.

Scenario

1. Antonio’s nurse convinced that using the app would be helpful and efficient for him and downloaded it on his smart phone.
2. Antonio’s nurse goes to the registration page and fills out all fields of the form (registration form contains some information such as: name, surname, gender, impairment, etc.) to create an account for Antonio.
3. The app welcomes the user with his name and goes to the main page.
4. Antonio receives a notification on his app about having an agenda and remembers that he needs to go to the physiotherapy room in 30 minutes.
5. Antonio gets ready and chooses the “Physiotherapy room” on his screen.
6. App gives suggestion to navigate the best way from his place to physiotherapy room.
7. Antonio follows the suggestions of the app step by step to reach the destination.
8. When he arrives to the destination, the App asks about his experience with the navigation system, Antonio rates the system and it will be sent to the App’s developers.

**Walking User**

Name & Family: Maria Rossetti

Age: 84

Disability: Visual impairment

I lost up to 40% of my kenning two years ago. From that time on I have some problem doing my personal things or walking around. So most of the times I need others help.

**Use of Technology**

Smart Phone: 50%  Tablet: 0%  PC: 20%

Maria is a sociable person who likes to participate in social and group activities and communicate with others, besides keeping her feeling of freedom.

**Scenario**

1. Maria’s nurse convinced that using the app would be helpful and efficient for her and downloaded it on her smart phone.
2. Maria’s nurse goes to the registration page and fills out all fields of the registration form to create an account for Maria (when in the registration page vision impairment is selected for Maria, the big font size is automatically set).
3. The app welcomes the user with her name and goes to the main page.
4. Maria decided to go to the restaurant to eat something.
5. Maria chooses the “Navigation” button in the main page of the App and then chooses restaurant from the list.
6. App guides her (in both ways of voice and graphic) to route the best way from her place to the restaurant.
7. She follows the suggestions of the app step by step to reach the destination.
8. When she arrives to the destination, the App asks about her experience with the navigation system, Maria rates the system and it will be sent to the App’s developers.

3.9. Deviation Scenarios:

According to our system, different kinds of deviation scenarios may happen during navigation. We define some of these scenarios in the following:

- **Delay Scenario:**
  
  Sometimes it is important that user reaches the destination room on an exact time. In our system this happens when user has an event in Agenda in which there is a due time for each event and user should be in a certain location on time and the arrival time is significant.

  Consider a user receiving a notification about attending a meeting. He has to be in the meeting room in at most 6 minutes. He starts moving towards the destination immediately and follows the App’s suggested path. But in the middle of his way in one of the corridors, he sees his friend and they start talking. He forgets about his meeting and keeps talking to his friend. He will be late for the meeting. He shouldn’t waste time during his way not more than the expected delay time considered by the system. So we need to notify him in some way.

- **Waste-Time Scenario:**
  
  In some other cases, unlike the previous scenario, the arrival time is less critical but we still need to make sure that user is moving through the path in a meaningful time period. In another word, we just want to make sure that user is moving through the path although in this case there is not any specific due time and we don’t care about the exact arrival time.

  For example, imagine a user sitting in his room, suddenly he decides to attend to a ceremony holding in the chapel, so he logins to the App and gets the direction to the chapel. But then he keeps turning around in his room to changes his cloths and takes his cane, etc. In this case, the user should be notified to prevent excessive tardiance. Actually we will
notify him to prevent staying in a room more than enough. So we may also need to consider the possibility of suspending/canceling the navigation by user’s desire, in this scenario.

- **Wrong-Orientation Scenario:**
  This scenario happens when user gets direction of the path but moves to the wrong direction or opposite direction of the suggested path, so in this case the moving orientation is wrong. This kind of deviation may happen everywhere in the path containing rooms; corridors, etc.
  Consider a user who uses the ALMA App to get direction to his destination and follows the path step by step to reach his destination. Somewhere in the path, he reaches a point in a corridor where the path is divided into two parts. The correct way is the left corridor, but he turns to the right corridor by mistake! Actually he turns to the opposite direction of the path. So again we need to use an algorithm to detect this kind of deviations and somehow notify him on time, to prevent further deviations.

- **Wrong-State Scenario:**
  In another scenario, consider a user who wants to get direction from his location to another place. He gets direction to his desired destination and follows the suggested path, in the middle of his way; he enters a room which contains several doors to other rooms. He absolutely should go to one specific room in order to stay in a correct path, but enters a wrong room by mistake! He definitely is on the wrong way and we should decide how to react. This kind of deviation may also happen when user is passing through the corridor and enters a wrong room in the middle of the path.
  If he continues this wrong direction he cannot reach the destination or may get lost in the building. So again we need an algorithm to diagnose these kinds of deviations and notify the user and in case it is needed calculate and show him a new path.
4. Design and implementation of Interface

The development and implement of the interface has been done in two phases:

4.1. Interface Design Phase

The first phase involves the design of a prototype taking into account the former interface and the requirements of the system. The goal of this phase is to achieve a common viewpoint about the appearance and layout of the interface (e.g. color theme, icons appearance, pages layout, etc.) with Alma system partners.

After perusing the former interface [7] (Figure 15 shows the some samples of the former interface), we found out that it has some imperfections which are listed below:

- Lack of attractiveness and eye-catching appearance (inappropriate colors and icons)
- Inconsistency in presentation and arrangement of the pages and icons

![Figure 15 Sample of the former interface pages](image)

Considering these disadvantages, we started to think about how to design a new Interface which can hold consistency, transparency and attractiveness of the application interface besides satisfying the requirements of the system. Then we started to prepare a prototype using InVision platform [8] (InVision is a prototyping tool which allows us to quickly and easily create interactive mockup for our designs).
When the design of the prototype prepared, we send it to our Alma system partners and it was satisfying for them. Figure 16 shows some examples of the prototype pages. Also the complete version of prototype is accessible through the following link:

https://invis.io/BQ34TUGCZ

Figure 16 Prototype presentation
4.2. Interface Implementation Phase

The second phase contains implementation and development of the interface, considering the prototype produced during the first phase and taking into account all the requirements of the system. In the following there is the implementation of the final interface design with some explanation about each page:

1. Login Page

This is the first page which users perceive, when try to open the Alma App. This page is being used to login to the system. In this page there are two buttons, “Log In” button for already registered users which leads the users to the main page of App and “Register” button which goes to the “Registration” page which is used to register new users to the system.

As you can see in Figure 17, several users can register and login to the system on a device. Also there are two different profiles which have different rights, the user “Nurse” acts as the admin of the system who can access more features and other users (Maria, Antonio) are the ordinary users of the system.

![Figure 17 Login page](image)
Nurses act as the administrator of the system and they have access to some secure information. So as you can see in Figure 18, entering to the Nurse users area is password protected.

![Figure 18 User Nurse Login page](image)

To make the use of application easier, we consider a “Remember” button in the login page. So if the user checks the "Remember" checkbox, the next time he/she enters the application will directly go to the main page. (Figure 19)

![Figure 19 Choose Remember checkbox](image)
2. Registration Page

In this page there are some fields embedded to insert user's information such as: name, surname, gender, kind of impairments, etc. By clicking on the “Register me” button the registration of the new user is complete and it leads to the main page of the application with a welcome message with user’s name. (Figure 20)

![Figure 20 Registration page](image1)

![Figure 21 Notification about filling out all the fields of registration page](image2)
All the fields of the registration page should be filled out, otherwise it is not possible to complete the registration and the system will notify the user to fill the empty field (Figure 21). If user has “vision” impairment, the system will automatically choose the big font size to be shown on the interface.

3. Main Page

In the main page of the application, users can choose what to do. On the top right of the page there is an icon which is used as “Log out” button and goes to the “Login” page.

There are four buttons at the center of the main page ” Agenda”, “Messaging”, “Navigation” and “Call Nurse” which lead the user to the related pages. (Figure 22)

![Figure 22 Main page](image)

4. Call page

By clicking on the “call nurse” button in the main page, a menu that asks the user if he is sure to make a call to the nurse appears. In this menu there is a possibility to confirm this action by clicking on “Yes” button or deny it by clicking on the “No” button. If user chooses “yes”, directly the nurse number will be dialed. (Figure 23)
5. Message Page

This page is used to have the possibility of sending a message to the nurse.

As you can see in Figure 24, there is a text field for typing a message body at the center of the page and the button “send message”. After clicking on the “send” button, user needs to confirm it. Also if user tries to send an empty message, system will notify him/her.
6. Agenda Page

This page shows user’s daily plan (Figure 25). By clicking on each of these plans, it will be redirected to the “Agenda detail” page which shows some information about that plan and also user can choose to get direction to the destination of that specific plan.

![Agenda page](image)

Figure 25 Agenda page

7. Agenda detail

This page shows Agenda plan’s details. At the center of this page some details about the selected plan is shown. By choosing the “Go” button user can go to Navigating page and get direction to the destination of the current plan (Figure 26).
8. Navigation Page

In this page user can see the list of his/her allowed destinations, clicking on each of them leads to the “Destination detail” page. (Figure 27)
9. Destination detail

This page shows some information about the selected destination in the navigation page. By choosing the “Go” button in the bottom of the page user can go to Navigating page and get direction to the selected destination (Figure 28).

![Destination detail page](image)

Figure 28 Destination detail page

10. Navigating Page

This page navigates the user to the selected destination (Figure 29). The arrow in the middle of the page shows the direction to the user.

At the bottom of this page there are some information about the passed steps, user speed and remaining distance to destination.
11. Feedback page
Within this page users will rate the navigation system, and they can also briefly explain about their experience (Figure 30 Feedback page).
12. Nurse area pages

The following pages are only accessible by user Nurse.

As it is shown in Figure 31, there is a list of all users in this page, so that nurse is able to edit/remove information of each user. Also at the top of this Page there is a menu bar consists of “Settings”, “Change Password” and “Info” options and beside the menu bar there is an icon which is used to logout of this page.

![Image of Nurse page]

*Figure 31* Nurse page

The setting page which is shown in Figure 32, gives the possibility of changing the App settings such as “language”, “font size”, “font color”, etc.
By clicking on the “Change password” option, a new menu appears which asks the user nurse to enter the old and the new password, so that the nurse user can change his/her password easily if needed (Figure 33).
Nurse has the possibility of editing target users’ information by clicking on the “Edit” button in front of each user's name. As it is shown in Figure 34, in this page nurse can change user's settings such as “Name”, “Surname”, “Impairment”, etc. and then save the changes by clicking on the tick icon. Also Nurse can remove a user by clicking on the trash icon.

As you can see in Figure 35, nurse is able to active/inactive call and message of the users. The result of inactivating call and message for user Maria is shown in right picture. After inactivating, the buttons of call and message are disabled and unclick-able for this user.
Figure 35 Active/Inactive call and message
5. Design of the Deviation Algorithms

In every navigation system, there is the possibility of being deviated during the path. If the system cannot detect deviation, users probably will be lost on their way. So we need to think about some accommodate and applicable ways in our system to diagnose and intercept possible deviations during navigation.

5.1. Deviation Analysis

- **Deviation Types**
  Deviation can happen in different conditions and different places such as rooms, corridors, etc. In any case, after analyzing and investigating the system, we came to the conclusion to define 4 kinds of deviations listed below:

  1. Delay Deviation
  2. Temporal Deviation
  3. Orientation Deviation
  4. State Deviation

  In the next section, Deviation Algorithms, we define and explain each of these algorithms in more details.

- **Deviation Notifications**
  In general, we have the following ways to notify the user when he deviates:

  1. Audio (beep or spoken words)
  2. Text (Pop-Up Box)
  3. Haptic (Vibration)
  4. Graphic (different color of arrow)

  Depending on the kind of deviation occurred and by the consideration of our users’ impairments; we can have the possibility to choose the ways to notify our users. For example, some users may have partly vision impairment, so we need to use Audio to notify them. Furthermore, some places and rooms may be so noisy so the user cannot hear the Audio notification or also some of our users may have hearing impairment, so we choose to embed Text notification as another way of notifying.
Also having vibration besides Audio and Text can attract user’s attention in case they did not notice the Audio and Text notifications.

So eventually, we decide to use the combination of Audio, Text and Vibration as the ways of notifications in our deviation algorithms.

5.2. Deviation Algorithms

5.2.1. Delay Deviation Algorithm

**Description:** The goal of this algorithm is to make sure that our user will get to their appointment’s destination on time, and if they are late we need to notify them in a way. Actually this algorithm happens when there is a due time on which user should be in the destination room.

So as it is shown in Figure 36, we compare the “Appointment time” minus “current time” with the “duration time from the current point of the path to the destination point” frequently. If it exceeds the expected duration time, we will notify the user, so user will be aware that he is going to be late. Whereas some destinations have top priorities, it is important to reach those destinations. For these destinations with high priority we decided not to give the possibility of exiting the navigation system, the user will just be notified about “being late”. For other destinations with low priority the possibility to exit the navigation is available, so in this case user is free to decide either to continue the path to reach the destination or to exit the navigation.

In fact the condition of this algorithm will be satisfied only once, and that is when the system should inform user.

**Notification:** We choose the combination of several kinds of notifications for this algorithm:

- Vibration
- Audio (Beep)
- Pop-Up text box: “You are going to be late, do you want to continue?”
5.2.2. Temporal Deviation

**Description:** In this algorithm, we try to avoid wasting time by user while he is getting direction of a path. Actually, we will inform him if he wastes time in one place more than approximate expected time “$t$”, which is the time we expect user reaches from his position to the next state of the path.

As we can see in Figure 37, first we get the current position of the user as state1 then wait for $t$ seconds and again check the current position of the user as state2. If it is still the same (state1=state2), it means that he is still in the same room and we need to notify him. This algorithm will do all these steps until the user reaches the destination. (In this algorithm by state we mean the states of the semantic layer)

**Calculation of time “$t$”:**

As we mentioned before $t$ is approximate expected time in which we expect that user reaches from state1 (User’s current state) to state2 (next state of the suggested path). The time parameter $t$ depends on two factors: “User’s speed” and “the distance between two states”.

We know that for each user there is an attribute “Mean Speed” on the server. So we can use it as $V_u$.

Also since we have the coordinates of each state, the distance “$d$” can be calculated using the following formulate:

**Coordinate of state1:** $(x_1, y_1)$

**Coordinate of state2:** $(x_2, y_2)$

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

So now by using the speed formula, we can obtain $T_u$:

$$T_u = \frac{d}{V_u}$$

And now we can calculate “$t$” for each user. Since we have different kinds of users and also different occurrences may happen, we cannot have the same time “$t$” for all of our users. So
by considering the following aspects, we can decide how much time to dedicate to each user:

- Our users have different kinds of steps and moving speeds (Short-Step-Slow, Short-Step-Fast, Normal-Step-Slow, Normal-Step-Fast)
- Users may need more time to open/close doors
- Users may have some mistake in the path
- Some unexpected factors may slow the user

So we define the following formula with different fractions to calculate “t”:

- Short-Step-Slow: \( t = \frac{5}{2} (T_u) \)
- Short-Step-Fast: \( t = 2 (T_u) \)
- Normal-Step-Slow: \( t = 2 (T_u) \)
- Normal-Step-Fast: \( t = \frac{3}{2} (T_u) \)

- The fraction of \( \frac{5}{2} \) is used for Short-Step-Slow users to give them 2.5 times more than expected time.
- The fraction of 2 is used for Short-Step-Fast and Normal-Step-Slow users to give them 2 times more than expected time.
- The fraction of \( \frac{3}{2} \) is used for Normal-Step-Slow users to give them 1.5 times more than expected time.

So in this case we can make sure that our users have a little more than expected time to move from one state to another.

(These fractions are adjustable through the interface on the setting page ("Time coefficient" field), so we can test their efficiency and in case it is needed it is possible to increase/decrease the fraction)

**Notification:** I choose the combination of following notifications for this kind of deviation:

- Vibration
- Audio: “Beep”
- Text box: “You are still in the same room; do you want to suspend the navigation?”
5.2.3. Orientation Deviation Algorithm

**Description:** The idea behind this algorithm is to make sure that the user is moving on the correct path suggested by the system.
We know that while user is moving on a correct path from his current state to the next state of the path, always the distance between these two states is reducing. So we can frequently calculate and check this distance and while it is reducing it means that user is on the correct path and is going to reach the next state of the path correctly. But if this distance increased, it means that user is deviated, so in this case we can notify him and tell him to return back to the correct path!

This idea gives users approximate freedom in their movement and to choose their path while navigating. So in case of existence of possible obstacle or crowed somewhere in path, they can choose to make a shortcut to the next state of the path. Since anyway the distance to the next state is reducing, he is moving on the correct path and would not be confused by over-notifying of the system.

As we can see in this algorithm (Figure 38), after starting the navigation, we frequently calculate the distance between current position of the user and the next state of the path and save it as $d_1$ and $d_2$ and compare these distances together. While $d_2$ is less than $d_1$ in means user is getting closer to the next state and is still moving on the correct path, otherwise ($d_2$ greater than $d_1$) means user is getting away from the path and system will notify him/her. This algorithm repeats until user reaches the destination.

(In this algorithm by state we mean the states of the navigation layer)

**Calculation of $d$:**

Since we have the coordinate of the states on the server, distance “$d$” is calculated using the following formula:

Coordinate of user’s current position: $(x_1,y_1)$
Coordinate of next state of the path: $(x_2,y_2)$

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$
In order to prevent the effect of feasible localization error on this algorithm, we consider adding a threshold to the comparison phase of \( d_1 \) and \( d_2 \), as follows:

\[(d_2 + \text{threshold}) < d_1\]

\[\text{threshold} = \text{max localization error}\]

And since the exact value of the max localization error is uncertain, we allocate a constant amount to it:

\[\text{Max localization error} = 2 \text{ meter}\]

**Notification:** I choose the combination of following kinds of notifications for this algorithm:

- Vibration
- Beep
- Pop-up text box: “Wrong Orientation”

### 5.2.4. State Deviation

**Description:** The idea behind this algorithm is to make sure that the user is correctly following the states of the suggested path. So as you can see in the algorithm (Figure 39), first we get all states of the suggested path in semantic layer, and then check the user’s position. If the current position of the user is not included to one of the states of the path, then we know that user is deviated from the suggested path and the system will inform him and then check if the user is back to the correct path or not. If user is not come back to the suggested path then will go to the “Re-calculate a new path” section. At this point we consider the new position of the user as starting point and request a new path to the destination. The new path would be either the shortest path from the point of deviation to the previously defined path or a new path to the destination

(As it is mentioned in the flowchart of ”Figure 39”, in this algorithm by state we mean the states of the semantic layer)

**Notification:** I choose the combination of following kinds of notifications for this algorithm:

- Vibration
- Beep
Pop-Up text box: “You are in a wrong cell!”

Figure 36 Delay deviation algorithm
Figure 37 Temporal deviation algorithm
Figure 38 Orientation deviation algorithm

- Get direction to the destination
- Calculate $d_1 = \text{distance between user's current position and the next state of the path}$
- Wait for a second
- Calculate $d_2 = \text{distance between user's current position and the next state of the path}$
- Check $d_2 < d_1$
  - Yes: Notify the User (You are going to the wrong direction, Please move through the suggested path!)
  - No: Check Destination reached?
    - Yes: END
    - No: Go back to Get direction to the destination

$\text{START}$

\begin{align*}
\text{d1} & = \text{distance between user's current position and the next state of the path} \\
\text{d2} & = \text{distance between user's current position and the next state of the path} \\
\text{d2 < d1} & \\
\text{Notify the User} & \text{ (You are going to the wrong direction, Please move through the suggested path!)} \\
\text{Destination reached?} & \\
\text{END} &
\end{align*}
Figure 39 State deviation algorithm
6. Implementation

6.1. Description of Classes

In this program “AlmaAddress”, “HttpManager”, “Requestpackage” and “ServerConnector” classes are used for connection to the server.

“AlmaAddress” class includes a list of different addresses for connecting to the server. There is server address, world address and users address in this class.

“HttpManager” class is used to get the detailed data from the server. In other words, it connects to the server using the “getData” method and receives the information of the requested URL. The “requestData” method is used to connect to api and in this function different methods such as get, post, etc. are defined as bellow:

```java
HttpURLConnection con = (HttpURLConnection) url.openConnection();
con.setRequestMethod("GET");
con.setRequestProperty("Accept", "application/json");
```

Then, this information is returned as a string.

The “requestDataPosition” method gets the values of x and y and connects to api and returns the state related to the position of x and y.

“Requestpackage” class is used to receive the main part of information in the server. This class is responsible for adjusting the URL. For example, “GetEncodeParam” method in this class returns the URL of states of a path. For instance, by receiving values of key and state and L0S21, it returns &start=L0S21 value.

“ServerConnector” class is used to recover, read and write data from the server. For example for receiving name, ID and agenda link of users this class is used. In general, the user information is recovered by this class. “getDestination” method returns the user’s destinations. “getUserState” method defines the current state of the user. “getUserPosition” method clarifies x, and y position of user and “getSpeed” method defines the user’s speed.

Agenda includes three main classes: “agenda”, “agendaArrayAdapter” and “agendaDetail”.

Agenda includes a list (listview) of user’s tasks. This list has been customized meaning that it consists of images and texts which is performed through agendaArrayAdapter class. After chosing a name in agenda list, its details such as explanation and execution time are...
displayed. (Using the agendaDetail class) after clicking on Go button, navigation starts (sensorActivity class)

Databases includes three classes: “DatabaseHelper”, “DBUserLoader”, “DatabaseHelper”. In this project database is used to save and recover user’s data.

“DatabaseHelper” class is used for the ease of use in relation to SQLite database and it consists of query to generate user table. To execute this command we use execSQL.

“DBAdapter” class is served to place the user information in database. “CreateUserContentValue” method presents different values such as name, surname, sex, etc. There are other methods like: “createUser” which creates new users in data base, “updateUser” which updates user info in database and “deleteUser” which deletes user from database.

“DBUserLoader” class is used to read the data from the server. In each execution of the application the user data is read from database and set for the user.

Destination includes three classes: “DestinationListAdapter”, “DestinationDetails”, and “Destination”.

“Destination” class includes of a list (listview) of user destinations. This list has been customized meaning that it consists of images and texts which is performed through “DestinationListAdapter” class. After choosing a name in destination list, its details such as explanation and execution time are displayed. (Using the “DestinationDetail” class) after clicking on Go button, navigation starts (“sensorActivity” class)

Management section consists of two classes: “ManagementActivity” and “ManagementArrayList”.

In “ManagementActivity” class user’s data is loaded from data base and displayed on the first page. To display user’s data, “ManagementArrayList” is used. This class provides the possibility of changing password, logging out from system, etc.

Map includes two classes: “path” and “point”.

“Path” class is used to: defining the distance between two points, getting information of a path such as state, geometry and expected_duration_sec.

“getDistanceBeforeTurn” method defines the distance between two points using latitude and longitude. The “getData” method, after receiving start and end points, it reaches the path information such as varies states to be passed. The “getDataL1” method, after
receiving start and end points, it reaches the path information in the L1 layer (semantic layer). The “getPoints” method defines the x, y position of different points. The “getStates” method returns the various states of the path. The “getStatesNames” method gets the name of states for voice class.

“point” class is used to describe the “point” variable. In this class we get or set the ID using the “getId” and “setId” methods. “setGeometry” and “getGeometry” methods define the x, y position.

Model includes three classes to describe objects: “AgendaObject”, “DestinationObject”, “SystemUser”.

“AgendaObject” is to define the agenda object. This object includes Title, Target, Date and Description. To get and set the agenda name the “getTitle” and “setTitle” methods are used. Likewise, the “getTarget” and “setTarget” methods are used to get and set the agenda destination. Same is for “getDate” and “setDate” methods which get and set the date. There are also “getDay” and “getDayName” methods which get the day and day name (Monday, Tuesday, etc.) respectively.

“DestinationObject” defines the destination object which includes ID and name. The two methods “getID” and “setID” are used to get and set the ID. The same is for “getName” and “setName” which are used to get and set the Name.

“systemUser” defines the user object and it holds some attributes like name, surname, sex, etc. “getName” and “setName” methods are used to get and set the name.

“SettingActivity” class is used to change some settings, for example font size, font color, language, etc. In this class the values which are stored in the memory are recalled.

“StepDetection” class determines whether the user has stepped or not. “detectData” method receives the data of accelerator sensor. The “getStepNum” and “setStepNum” functions get and set the number of steps respectively.

Main activities of the program

“InfoActivity” This is used to display program information.

“LoginActivity” is displayed as soon as user logs in. The “loadSpinnerData” method places the data in login page spinner. The “isOnLine” method returns the true value when user is online. The “showNoConnectionDialog” method shows this dialog if there is no connection.

“MainActivity” is used to display the main page containing four main segments of the program. This class connects to server and gets user info. “showCallDialog” method is used to display the call window. The user can call the nurse through this dialog.
“MessagingActivity” class is responsible for sending message to the nurse. The user can send text message to the nurse using this class and he/she is asked to confirm sending the message.

“RegisterActivity” activity is used to register a user. To register the user various instances are asked such as name, surname, birth date, impairment type, sex, height and step type. At the end these values are saved in the data base.

“SensorActivity” activity includes sensors as well as navigation algorithms. Using this activity, the user can find the appropriate path.

6.2. Method of getting user’s data
At first, in the “MainActivity” class define an object of “SystemUsers” and call this object “user”. User’s data are received by executing a private subclass “MyTask” in “MainActivity”. In “MyTask” user’s address link is built as follows:

AlmaAdresses.Users + userId + ".js";

The mentioned address for user 1 is as follows:

http://isin12.dti.supsi.ch:5001/worlds/ComoSim/users/1.js

by using “getData” method in “HttpManager” class, the contents of this link will be get and store in a json object. Then through “ServerConnector” user’s data such as Agenda, Destinations, Speed, State and Geometry are assigned.

6.3. Method of getting Agenda

In Agenda, the user’s agenda data that we get upon user’s data will be used and agenda details such as title, time and descriptions is extracted. “getAgenda” method in “serverConnector” class is used to the agenda details. This method gets the user’s ID and connects to the following link:

AlmaAdresses.Users + userid + "/agenda";

the mentioned address for user 1 is as follows:
Agenda notification transmission

“startNotification” method in “UpdateService” class is implemented for this purpose. This method checks two conditions frequently:

- if this is the morning of an agenda
- if there is 5 minutes left to an agenda

If any of the above conditions are satisfied, sends the notification to the user’s device. So every morning user will receive notification about the agenda of the day. Also another notification will be send to the user 5 minutes before each agenda plan.

6.4. Method of getting path

Path information is got by using “myTask” subclass in “sensorActivity” class. In a way that, start and end points are given to this class, then by calling the method “getData” (which is defined in “path” class) we can get path information.

“getData” method sends the following link to api: 
AlmaAdresses.SERVER + AlmaAdresses.COMO + "map/path"

For example, if the source point is “L0S36” and the destination point is “L0S1”, the address is like this:
GET /worlds/ComoSim/map/path?start=L0S36&end=L0S1

As it is shown “GET” is used to receive the data and then start and end points are added to the end at the end of the address.

The output is returned as a json file from the server which contains all the states of the path and their properties.

6.5. Deviation algorithms

Deviation detection algorithms are implemented within the “SensorActivity” class and it is needed to connect to the server to implement them, if the connection to the server be accomplished directly, because of internet disorders some errors may occur. To prevent these errors we use Async. AsyncTask is an abstract Android class which helps the Android
applications to handle the Main UI thread in efficient way. AsyncTask class allows us to perform long lasting tasks/background operations and show the result on the UI thread without affecting the main thread [9].

All the deviation algorithms are defined in “SensorActivity” class as private subclasses and in the async format. First, by connecting to api, the path and the states of the path to a specific destination are determined. Then this information is stored and the path is updated. These information is needed for all deviation algorithms and it is implemented in “MyTask” subclass (This section directly uses the “Path” class).

“getStateNames” subclass is used for getting state’s names in Semantic layer L1. “getStatesNames” subclass uses “getStatesNames” method in “Path” class to get state’s names and these will be used in voice-guide and State deviation algorithm.

The following links are for connecting to the server and getting user’s data:

```java
String userUrl = AlmaAdresses.Users + MainActivity.userId + ".js";
String content = HttpManager.getData(userUrl);
```

The above link is as follows for the first user:

```
http://isin12.dti.supsi.ch:5001/worlds/ComoSim/users/1.js
```

### 6.5.1. Delay-Deviation implementation

This algorithm is implemented within “DelayDeviation” subclass. This algorithm executes only if the user is getting a path through Agenda. In the agenda, the plan time is specified. To get the agenda, we use “getAgenda” method in “serverConnector”. This method gets user’s ID and connects to the following link:

```
AlmaAdresses.Users + userId + "/agenda";
```

This address is as follows for the first user:

```
http://isin12.dti.supsi.ch:5001/worlds/ComoSim/users/1/agenda
```

This algorithm regularly checks if there is enough time to reach the destination position (already explained in section 5.2.1). If the selected plan has low priority, user will be notified in a dialog box and can choose to exit the navigation, but if the plan has high priority, user will just be notified via toast pop-up notification (Figure 40).
6.5.2. Temporal-Deviation implementation

This algorithm is implemented within “TemporalDeviation” subclass. In this algorithm, first we connect to the server and get user’s data. For this purpose the following link is used:

```java
String userUrl = AlmaAdresses.Users + MainActivity.userId + ".js";
String content = HttpManager.getData(userUrl);
```

This link for the first user is as follows:

```
http://isin12.dti.supsi.ch:5001/worlds/ComoSim/users/1.js
```

So now we can check user’s position regularly (every t seconds, already explained the calculation of t in section 5.2.2) and compare with his former position and in case the user’s position is the same it means that user is still in the same room and he/she can choose to suspend the navigation or to continue the path (Figure 41).
6.5.3. Orientation-Deviation Implementation

This algorithm is implemented within “OrientationDeviation” subclass. The same as the previous algorithm, again we connect to the server and get user’s data using the following link is used:

```java
String userUrl = AlmaAdresses.Users + MainActivity.userId + ".js";
String content = HttpManager.getData(userUrl);
```

For this algorithm, we need x, y position of the user to check the distance between user and next state frequently. If this distance increases in comparison with the former distance, it means the user is getting away from the path and will receive notification (Figure 42).
6.5.4. State-Deviation Implementation

This algorithm is defined within “StateDeviation” subclass. For this algorithm we use data of subclass “getStatesName” in which “getStatesName” and “getData” methods in “path” class are called. “getData” method first sends the following link to api:

AlmaAdresses.SERVER + AlmaAdresses.COMO + "map/path"

For example, with the start point of “L0S36” and the end point of “L0S1” the address is like this:

GET /worlds/ComoSim/map/path?start=L0S36&end=L0S1

As you can see, GET is used to receive data and then “start” and “end” parameters are added to the end of the link. This value is stored in “myStateName” array.

In each execution of the algorithm, we get the x and y position of the user. Then again connect to the server to convert x, y position of the user to its related state. If the obtained state does not include in the states of the path, the notification text is shown (Figure 43).

Figure 43 State-Deviation notification
7. Conclusions and Future Work

Within this project we implemented and developed a user-friendly interface application of the Alma system, through which users can access to different services such as:

- indoor navigation assistant, through which users can get direction to the selected destination
- List of agenda of plans, through which users can check their daily events, receive notification about each plan every day and half an hour before it and also get direction to destination of that plan
- Call service, so that users can make direct calls to their nurse in case they need help or have a request
- Message service, using which users can send message directly to their nurse

In addition four new innovative deviation algorithms have been introduced and implemented categorized to:

- Delay-Deviation Algorithm: to make sure that user reaches his/her appointment’s destination on time, in case of being late, he/she will be notified. User can choose to exit the navigation through this algorithm, if the priority of the selected plan in low.

- Temporal-Deviation Algorithm: to diagnose and intercepts if user waste time during a path. User can choose to suspend the navigation through this algorithm

- Orientation-Deviation Algorithm: which detects and intercepts if the user is moving towards a wrong direction according to the suggested path

- State-Deviation Algorithm: which detects and intercepts if the user is in a wrong state regarding to the states of the suggested path

As a future work, the navigation system can be improved by improving the localization module with the possibility of detecting different floors.
Also it is suggested to reconsider the requirements of the system for users with harder or different kind of impairments (e.g. blind users) and accordingly improve and develop PNA so that it can be used by more widespread range of users.
Bibliography

2 http://atg.deib.polimi.it/projects/
3 http://indoorgml.net/
4 https://www.politesi.polimi.it/bitstream/10589/94921/1/2014_10_IMANI.pdf
6 https://developer.android.com/about/dashboards/index.html
7 https://www.politesi.polimi.it/bitstream/10589/94921/1/2014_10_IMANI.pdf
8 https://www.invisionapp.com/?ent_boxes=true&sticky=true
9 http://programmerguru.com/android-tutorial/what-is-asynctask-in-android/