



Project Acronym: **BIMERR**  
 Project Full Title: **BIM-based holistic tools for Energy-driven Renovation of existing Residences**  
 Grant Agreement: **820621**  
 Project Duration: **45 months**

## DELIVERABLE D6.10

### Renovation progress monitoring & alerting application for residents 1

Deliverable Status: **Final**  
 File Name: **Deliverable D6.10**  
 Due Date: **30/07/2020 (M19)**  
 Submission Date: **21/08/2020 (M20)**  
 Task Leader: **NT (T6.5)**

Dissemination level	
Public	X
Confidential, only for members of the Consortium (including the Commission Services)	



This project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°820621

### **Disclaimer**

#### **The project consortium is composed of:**

FIT	Fraunhofer Gesellschaft Zur Foerderung Der Angewandten Forschung E.V.	Germany
CERTH	Ethniko Kentro Erevnas Kai Technologikis Anaptyxis	Greece
UPM	Universidad Politecnica De Madrid	Spain
UBITECH	Ubitech Limited	Cyprus
SUITE5	Suite5 Data Intelligence Solutions Limited	Cyprus
HYPERTECH	Hypertech (Chaipertek) Anonymos Viomichaniki Emporiki Etaireia Pliroforikis Kai Neon Technologion	Greece
MERIT	Merit Consulting House Sprl	Belgium
XYLEM	Xylem Science And Technology Management Gmbh	Austria
CONKAT	Anonymos Etaireia Kataskevon Technikon Ergon, Emporikon Viomichanikonkai Nautiliakon Epicheiriseon Kon'kat	Greece
BOC	Boc Asset Management Gmbh	Austria
BX	Budimex Sa	Poland
UOP	University Of Peloponnese	Greece
UCL	Univeristy College London	United Kingdom
UEDIN	University of Edinburgh	United Kingdom
NT	Novitech As	Slovakia
FER	Ferrovial Agroman S.A	Spain

*BIMERR project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°820621. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Commission (EC). EC is not liable for any use that may be made of the information contained therein.*

## AUTHORS LIST

Leading Author (Editor)				
	Surname	First Name	Beneficiary	Contact email
	Tsakiris	Athanasios	CERTH	atsakir@iti.gr
Co-authors (in alphabetic order)				
#	Surname	First Name	Beneficiary	Contact email
1	Andriopoulos	Athanasios	CERTH	aandriop@iti.gr
2	Tsita	Anastasia	CERTH	a.tsita@iti.gr
3	Pantraki	Evanglia	CERTH	epantrak@iti.gr
4	Sfikas	Georgios	CERTH	gsfikas@iti.gr

## REVIEWERS LIST

List of Reviewers (in alphabetic order)				
#	Surname	First Name	Beneficiary	Contact email
1	Demeter	Dominik	NT	demeter@novitechgroup.sk
2	Vergeti	Danai	UBITECH	vergetid@ubitech.eu

## REVISION CONTROL

Version	Author	Date	Status
0.1	CERTH	25/05/2020	ToC
0.5	CERTH	27/06/2020	Ready for 1 <sup>st</sup> Internal Review
0.6	CERTH	18/07/2020	Ready for 2 <sup>nd</sup> Internal Review
0.9	CERTH	28/07/2020	Ready for peer review
1.0	CERTH	20/08/2020	Ready for submission

## TABLE OF CONTENTS

<b>List of Figures .....</b>	<b>8</b>
<b>Acronyms .....</b>	<b>10</b>
<b>Executive Summary .....</b>	<b>11</b>
<b>1. Introduction.....</b>	<b>12</b>
1.1. Objectives of the Deliverable.....	12
1.2. Introduction of Design, Methodology and Implementation .....	12
1.2.1. Design Process.....	12
1.2.2. User Requirements and Use Cases.....	13
1.2.3. Development plan and Technical Requirements .....	13
1.2.4. Implementation, Features and Application Workflow .....	14
1.3. Structure of the Deliverable.....	14
<b>2. Design Process.....</b>	<b>16</b>
2.1. Introduction .....	16
2.2. PWMA Application for Residents' main goal and functionalities.....	16
2.3. GUI Design Approach .....	21
2.4. GUI Design Methodology .....	22
<b>3. User Requirements and Use Cases.....</b>	<b>25</b>
3.1. Introduction .....	25

<b>3.2. Occupants' requirements .....</b>	<b>26</b>
3.2.1. Requirements.....	26
3.2.2. Personas and use case stories.....	27
3.2.3. Information Architecture & Navigation .....	32
3.2.4. Design Choices - Layout .....	37
<b>3.3. Use cases.....</b>	<b>41</b>
3.3.1. UC – 01: Notifications and residents' responses related to tasks.....	41
3.3.2. UC – 02: Residents' responses related to a specific task .....	42
3.3.3. UC – 03: Residents' new issue reports.....	42
<b>4. Development plan and Technical Requirements .....</b>	<b>44</b>
<b>4.1 Development Plan .....</b>	<b>44</b>
<b>4.2 Development Toolset.....</b>	<b>45</b>
4.2.1 Unity.....	45
2.2.2 C#.....	46
2.2.3 Visual studio.....	46
<b>3 Implementation, Features and Application Workflow.....</b>	<b>47</b>
<b>5.1 Implementation.....</b>	<b>47</b>
<b>5.2 Features .....</b>	<b>49</b>
<b>5.3 Application Workflow .....</b>	<b>61</b>
<b>4 Next Steps .....</b>	<b>67</b>

<b>5</b>	<b><i>Conclusion and Outlook .....</i></b>	<b>69</b>
<b>6</b>	<b><i>References.....</i></b>	<b>70</b>

## LIST OF FIGURES

Figure 1 PWMA Application for residents General Architecture.....	14
Figure 2 Iterative Design & Development Methodology .....	23
Figure 3 Iteration circle of Design & Development Methodology.....	24
Figure 4 Home page “Tasks” - List view (left), Calendar view (center), and Menu (Navigation Drawer) (right) .....	32
Figure 5 Information Architecture .....	35
Figure 6 Navigation Diagram – Mock-up screens .....	36
Figure 7 Mock-ups, Colors and Indicators Home page “Tasks” - List view (left), Calendar view (right) .....	37
Figure 8 Mock-ups, Android Notifications (left), “Notifications” page (center), “Settings”- Notifications Settings (right).....	39
Figure 9 Mock-Ups, Selected task – Details of tasks (second navigation level).....	40
Figure 10 Mock-Ups, “New Issue” page: Location confirmation (Left), Type of issue selection (Center), Add Description (Right).....	41
Figure 11 Unity environment.....	46
Figure 12 Example of triggering the Main Menu.....	48
Figure 13 Example of the GUI which remains relative in every aspect ratio.....	49
Figure 14 Login screen.....	50
Figure 15 Register screen (left), pop up window for finalizing registration (right).....	51



Figure 16 Task tab – list view .....	52
Figure 17 Selected task – Task Details, into the second navigation level.....	53
Figure 18 Tasks Tab – Calendar View .....	54
Figure 19 "New Issue" tab: Location confirmation.....	55
Figure 20 "New Issue" tab: Type of issue selection.....	56
Figure 21 "New Issue" tab: Add Description.....	57
Figure 22 Android Notifications (Left), In App Notifications (Right) .....	59
Figure 23 Settings - Notification Settings.....	60
Figure 24 Menu (Navigation Drawer) .....	61
Figure 25 Retrieved JSON file from PWMA REST API.....	62
Figure 26 JSON serialization .....	63
Figure 27 Parsing through JSON Utility.....	63
Figure 28 Application workflow.....	66

## ACRONYMS

Acronym	Meaning
AEC	Architecture Engineering Construction
API	Application Programming Interface
ARIBFA	Augmented Reality enabled In-situ Building Feature Annotation
BICA	Building Information Collection Application
BIF	BIMERR Interoperability Framework
BIM	Building Information Management
BIMERR	BIM-based holistic tools for Energy-driven Renovation of existing Residences
BISP	Building Information Secure Provisioning
FAQ	Frequently Asked Question
GUI	Graphical User Interface
H&S	Health and Safety
HTTPS	HyperText Transfer Protocol Secure
IoT	Internet of things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
PWMA	Process & Workflow Modelling & Automation
REST	Representational State Transfer
UC	Use Case
WCAG	Web Content Accessibility Guidelines

## **EXECUTIVE SUMMARY**

This document analyzes the first version of the mobile application for on-site support of the residents and corresponds to the approach that is described in Deliverable 3.1. The application focuses on creating a direct communication channel between building managers and residents, which can result in enhanced quality of renovation services and the implementation of formative evaluation as input in planning renovation processes, in Architecture Engineering Construction (AEC) projects.

# **1. INTRODUCTION**

## **1.1. OBJECTIVES OF THE DELIVERABLE**

This deliverable introduces the first version of the mobile application for on-site support of the residents and corresponds to the approach that is described in Deliverable 3.1. The purpose is to achieve a direct communication between building managers and residents to enhance the quality of renovation services and implement formative evaluation as input in planning renovation processes from people that use the space on a daily basis, in Architecture Engineering Construction (AEC) projects. The provided application is part of a standalone toolset based on the Process & Workflow Modelling & Automation (PWMA) platform adjusted for the needs of BIMERR.

The first version of the application aims at providing a service that properly allows interactions with the tasks assigned to the user. The next iteration due for M30 will be integrated more deeply into the overall BIMERR ecosystem, especially in terms of utilization of the common Authentication mechanism as well as integrated cooperation with the notification system.

## **1.2. INTRODUCTION OF DESIGN, METHODOLOGY AND IMPLEMENTATION**

There are several different methodologies to follow in the development of an end-user oriented application each one with different features and steps. In the case of the Residents' application for PWMA, the decision was made to follow a user-centered design approach that covers the different phases of the design from the scope of the user needs and requirements, involves the end-user in the definition of the use cases and features required from the application and verifies the design process through user participation and verification of requirements met. This approach is encompassed with an iteration-based implementation cycle where the end-users are involved in each iteration to test and validate the end result of each design phase.

### **1.2.1. Design Process**

Every design process starts with the definition of the problems that the final product, interactive system or service is about to solve. User and system requirements (design constraints, functional requirements etc.) are composed of both the analysed information on end-users needs arise from structure or semi-structure user requirements research methods (interviews, questionnaires, brainstorming etc.) and

technical specifications arouse mainly from the system architecture proposed within the scope of the project and each task goals. Proposed solutions are first revised through possible flows of user actions and interface visualizations using a variety of means such as personas-based scenarios, wireframing, and mock-up visualizations of main interactive graphical interface components. Mock – ups can be used to feed the development step of the application. Feedback, depending on the stage of development can be gathered through discussions and crosschecking between designers, developers and stakeholders, or, if in final stages, through tests with end-users combined with formative evaluation tools (e.g. questionnaires) based on near real-life situated and pilot specific use cases. In the frame of PWMA application user-interface design, an iterative and incremental process is followed, with the first iteration resulting in mock-ups that feeds the development of the first version of the application. Next iterations incorporate that feedback to expand the application development both in detail and content, but also to revise previously made design choices.

### **1.2.2. User Requirements and Use Cases**

The User Requirements specify what users expect from the system in terms of its functionalities. User Requirements were written before the system is created. The results from the questionnaires analysis are mapped onto BIMERR user requirements. The prioritization of end users requirements is also produced, setting the initial guidelines for the design and development of the App for residents. [D3.1 stakeholder]

Use case is a list of actions defining the interactions between a role (an actor) and a system for achieving a goal. Based on the proposed Business Scenarios, use cases are identified. The aim is to describe a list of system use cases that highlight the main principles of the App for residents and hence, an appropriate subset can be tested during the pilot trials. The use cases are defined after consultation both with the technical and the pilot partners of the project.

### **1.2.3. Development plan and Technical Requirements**

After the design process is finished and the resulting mock ups are at hand, the next phase was to plan the implementation process. This process is based on the initial development plan and technical requirements. This stage consists of three parts: the selection of the methodology, the creation of the timetable and the research process along with the determination of the application's architecture. The

architecture covers the general functionalities of the application as well as the "ecosystem" of external sources with which information will be exchanged during the project. The general structure of the application's architecture is presented in Figure 1.

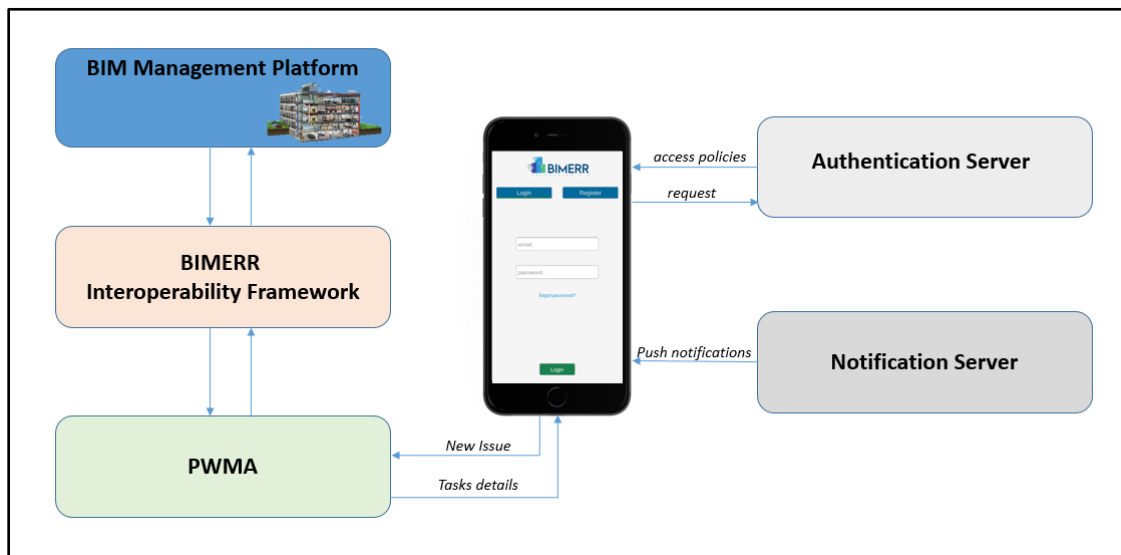


Figure 1 PWMA Application for residents General Architecture

#### 1.2.4. Implementation, Features and Application Workflow

The implementation of the application begins when all of the previous steps have been successfully completed. Implementation is the carrying out, execution of the development plan. The results of the implementation provide features to the user of the app. In Chapter 5, the implementation and the features of App for residents will be described, as well as the workflow process which is a series of sequential tasks that are carried out based on user-defined rules or conditions, that govern the operation of the application. The workflow shows front-end and back-end process alongside.

### 1.3. STRUCTURE OF THE DELIVERABLE

Section 2.2 presents the path towards the definition of the main goal and main functions of the application. Use cases were presented in Deliverable 3.1 and functionalities regarding Building Occupant user group interfaces were split to form both the Building Information Collection Application (BICA) and

the Process & Workflow Modelling & Automation (PWMA), based on the list of occupant user requirements (requirements BMRR-151 – 163). The design process starts with a clear statement of the main goal of the application and the main functions stated hierarchically. Not major changes are expected to be made on those in the next steps of the application's development.

In the section 2.3, the general design approach is described shortly to depict the theoretical base. It is decided that material design framework [1] will be the base for all BIMERR mobile applications, as a highly flexible and documented framework that successfully proposes and incorporates major usability guidelines and can be aligned with universal design theoretical approach.

In the section 2.4, the design methodology is described. It is an iterative and incremental approach, mainly due to the high dependency of the application on other modules of the project and the dynamic nature of workflow and human interaction involved.

User requirements and use cases are described in chapter 3 and note the beginning of the first iteration, which closed with a version of mock-ups that lead to a first version of the application developed. Every new iteration circle starts with re-defining requirements and ends with a next version of the application developed. Feedback on one circle triggers and reforms the goals for the next circle of design and further development.

In chapter 4 the development plan and methodology are described. Moreover, technical requirements are justified based on the tools that were used for the development.

In chapter 5 the implementation is described, including the interactions, tools, and components that were used to develop the features given to the users and the application's workflow between back and front end.

Chapters 6 and 7 summarize and conclude the next steps in design and development process.

## **2. DESIGN PROCESS**

### **2.1. INTRODUCTION**

The Graphical User Interface (GUI) design process starts at the clear statement of the main goal and main functionalities of the PWMA application (section 2.2.), which was an outcome of BIMERR requirements (Deliverable 3.1.) and discussion among the project's partners. The design process that follows is based on the theoretical approach of universal design theory and its main principles, towards an inclusive design for all occupants, regardless their abilities and knowledge, their experiences and cultural background. All BIMERR applications were decided to follow material design framework and guidelines for both visual and functional consistency. Material design framework and guidelines inherently embed inclusive and universal design principles. The methodology follows an iterative and incremental design process. Feedback on a design circle highlights gaps and inconsistencies that feeds a new circle, which starts with re-defining and prioritizing solutions for the next design choices towards a more detailed and coherent GUI.

Personas were used to depict the users' needs and requirements based on the user requirements of the PWMA application (Section 3.2.1.), in order to better understand the context of use and the specific story-based problems of the users that the application aspires to solve. This way the flow of actions and the components chosen for the visualization of the main screens of the application was possible, and specific design choices were made regarding the features of the application. The functionalities per page were hierarchically organized in the information architecture diagram. Mock-Ups of the main screens and the navigation diagram were the final deliverables to the development team, discussed and further elaborated at the beginning and during the first development process.

### **2.2. PWMA APPLICATION FOR RESIDENTS' MAIN GOAL AND FUNCTIONALITIES**

In order to understand the services that will actively involve the residents, and examine their ways and level of participation in applications' functionalities, within the scope of the project, the PWMA team focused on the use cases that involve the user group Building Occupants.

The following tables include those use cases that involve the Building Occupants user group. The aspects of the use case description that refers to the involvement and contribution of that user group, as well as



the questions aroused on those aspects under the spectrum of the applications' functionalities are noted for every BIMERR use case.

**Notes on residents' involvement and contribution based on the use cases:**

<b>BIMERR Use Case 2</b>	
UC-02	Accelerate the collection of data about the building systems through BIM-based internal audit support tools and interaction with building managers and occupants.
Building Occupant related aspects	The occupants of the building having better knowledge of their building will be able to upload supplementary data to the BIM model, utilizing an easy to use application. All gathered information will be used to enrich the pre-designed BIM model, accelerating the overall collection of data required for the initial renovation scenario modelling process and contribute to the avoidance of critical delays or extra investments.
Notes	<ul style="list-style-type: none"> <li>• The residents input is important, as at times they have irreplaceable knowledge.</li> <li>• The resident cannot directly annotate or comment elements from the BIM model. A filter is probably needed, to take the most relevant information and then incorporate it in the BIM model.</li> <li>• Input directly on BIM would be a more professional tool, and thus not easy or meaningful to be used by everyone.</li> </ul>

<b>BIMERR Use Case 3</b>	
UC-03	Adapt design to the actual building use, including accurate information about occupancy and schedules, comfort requirements/ preferences and energy uses.
Building Occupant related aspects	Data from IoT devices installed in the building to be renovated will be made available to a profiling tool which will perform accurate profiling of building use aspects to generate detailed occupancy and behavior profiles of occupants (energy consumption, behavioral patterns, building usage etc.).
Notes	<ul style="list-style-type: none"> <li>• Not all residents, under the view of a more generic application of the project, would have IoT devices installed in the spaces they use or not with the same profile structure. As participation of residents is important, maybe there should be a more independent solution. For example, an application that would gather information on residents' needs, that they would input manually and at any time as they wish.</li> </ul>

## **BIMERR Use Case 10**

UC-10	Continuous reporting from workforce and occupants for changes performed over the initial renovation design (location-based on a BIM representation) and automated update of the BIM model (as-built documentation).
Building Occupant related aspects	Workers and occupants/owners are in need of easy-to-use applications to report changes in the renovation project during the construction, in real-time. Occupants will use appropriate tools that will allow them through their smart phone to verify the compliance of the works performed against the original plans and provide feedback on occurred changes. Occupants, apart from verification they will have the opportunity to introduce proposals.
Notes	<ul style="list-style-type: none"> <li>• Maybe not all occupants would be interested to get involved or fully understand the works performed and why. Occupants are not always owners.</li> <li>• Verification on compliance of the works performed against the original plans maybe at times not be clear to an occupant with not specialized knowledge. The verification should be optional for an occupant, and, ideally, should be done with or without the participation of occupants. Someone from the renovation stakeholders should verify anyway.</li> <li>• Regarding the feedback and the introduction of slight changes from the occupants should be expressed through a simple way and as they perceive it.</li> </ul>

BIMERR Use Case 11	
UC-11	Identification of threats and dangers and provision of alerts to workforce and occupants through BIM-based apps and UIs
Building Occupant related aspects	Any relevant issue that may have a negative impact to in situ H&S needs to be promptly reported, documented and transparently communicated to all involved actors. Occupants/owners plan to visit the construction site. Before entering the construction site, they use an available app to display H&S instructions that should be followed during their visit. Moreover, for visitors and workers, potential risks could be displayed for each floor of the building.
Notes	<ul style="list-style-type: none"> <li>• Health and Safety (H&amp;S) issues should always reach all residents of the building that will be or is under renovation. All residents should be notified for every inconvenience (e.g. water cut) and for every related schedule change that might happen during works. Instructions and recommendations should be expressed with simple terms.</li> </ul>

BIMERR Use Case 12	
UC-12	Continuous reporting from workforce and occupants for dangers and threats (location-based on a BIM representation) and automated update of the BIM model.
Building Occupant	Apart from possible H&S issues identified and highlighted by the H&S manager, there might be relevant issues that come up during the actual work performed on-site. Such issues may be known only to workers or observed by occupants visiting the renovation site, who shall

related aspects	make this information available to any other actor involved to avoid possible accidents and safety-related incidents. The occupant creates an H&S issue by using the relevant user interface. The H&S manager receives notification about the problem and its location. Once the problem is solved, the H&S manager changes the issue status. Accurate and prompt notifications to involved actors for H&S issues to prevent against severe incidents at the construction site.
<b>Notes</b>	<ul style="list-style-type: none"> <li>• Occupants use the space in a daily base, thus they might be the only ones to observe issues as they use the space even after renovation is completed.</li> <li>• They should be able to report any issue and at any time through a simple and easy-to-use interface.</li> </ul>

<b>BIMERR Use Case 13</b>	
UC-13	Perform back-to-back simulations of alternative renovation scenarios to evaluate and select the best energy-performing renovation scenario
Building Occupant related aspects	The tool will utilize valuable information regarding energy behavior of the building occupants, together with cost elements for different materials extracted from a complete database that is integrated with the building energy performance simulation module.
<b>Notes</b>	<ul style="list-style-type: none"> <li>• As also noted in use case 3, not all residents would have IoT devices installed in the spaces they use or not with the same profile structure. All of them should have the opportunity to contribute with their concerns and feedback.</li> </ul>

<b>BIMERR Use Case 15</b>	
UC-15	Energy performance simulations to assess not only energy metrics, but also accurately evaluate occupants' comfort and indoor air quality
Building Occupant related aspects	Introduction of indoor conditions (temperature, humidity, surface temperature) as another metric to evaluate the effectiveness of the renovation scenarios is key to the real estate value enhancement of renovation projects. In this context architects needs to have access to detailed comfort profiles and hygienic/ health requirements of occupants in buildings, which need to be inferred in a non-intrusive manner. Subsequently, such profiles and boundaries need to be introduced in a powerful renovation decision support tool to address both these aspects in any simulation performed and corresponding evaluations.
<b>Notes</b>	<ul style="list-style-type: none"> <li>• As also noted in use cases 3 and 13, not all residents would have IoT devices installed in the spaces they use or not with the same profile structure. All of them should have the opportunity to contribute with their concerns and feedback.</li> </ul>

All in all, PWMA application addresses every resident of renovation site, who would not need expert knowledge on problem diagnostics or identifying missing points in Building Information Model (BIM). This application would not necessarily give access to the BIM model for the residents or to detailed renovations plans and would be simple enough to encourage “everyone” to get involved and interested in the space they use, including residents that do not have IoT devices installed in their space (energy efficiency, comfort zone, behavior, schedules etc.). PWMA application for residents, swifts towards including as many residents as possible and invite them into the process of renovation planning but does not ask much on their opinion neither requires any specialized knowledge on specific renovation plans or specific BIM building components. The goal is to mostly focus on notifying all residents about upcoming and ongoing planned renovation works that affect their building or area (H&S issues), while allowing them to “reach” renovation managers and stakeholders and communicate their needs and concerns, creating an open dialog mainly on the use of space, and not that much on the space itself. Finally, a simple application, that would encourage participation by all, seems as an opportunity to utilize the feedback from everyone that uses the space in a daily base. For example, it is fact that some issues might be observed even months after a renovation work completion and only by use. This feedback would be valuable for renovation services enhancement in both current and future works.

In contrast to the purpose of the PWMA application for residents where the use cases described above require the app to provide information to the user regarding the processes and tasks performed during the renovation, the main goal of BICA application is to collect additional building information from the residents/owners of a renovation site prior to the actual construction. The users participate in diagnostic procedures, identify and verify interventions to the BIM, and have access to details about the building equipment/components. Through the profiling tool and IoT devices they share information regarding their comfort status at various environmental conditions, but also their preferred use of the spaces and components.

**PWMA application goal statement:** Mobile application for resident inclusion in AEC projects, towards a direct communication with residents to avoid on-site accidents and ensure H&S for occupants, to enhance the quality of renovation services and implement formative evaluation as input in planning renovation processes from people that use the space in a daily base.

#### **PWMA application main Functionalities (hierarchically):**

- 1) The main function is to notify residents for scheduled renovation tasks in their area (building, apartment) and the effects of them in health and safety issues, such as planned electricity and water cuts, high noise levels, facilities or spaces out of order, dangers etc. The users should be

able to follow up with the scheduled tasks and their effects at their own time regardless of receiving notifications or not.

- 2) The residents will be included in the process of AEC projects and their formative and final evaluation, given the opportunity to reply to those notifications and add their comments for a pending or on going task, as well as confirm whether a task has been completed successfully or not.
- 3) Additionally they will have the opportunity to report any issue they encounter during their everyday use of space, regarding malfunctions, suggestions of improvements etc. at any time regardless of an ongoing renovation work.

### **2.3. GUI DESIGN APPROACH**

The application is targeting “all”, given the fact that residents are people of all ages, with different levels of confidence in use of technology, with different educational level and abilities, or living arrangements (family, alone etc.), different experiences and cultural background. Universal design theoretical approach, inherently proposes a flexible, inclusive design centered around accessibility for both disabled and nondisabled users. The outcome of such design approach, ideally, benefits “everyone”, or most people, who are given equal access to all features of a product and in the same or similar way, without stigmatizing any novice, elder, or disabled user [2],[3],[4].

Following a universal design approach for the application interface layout, look, navigation, content arrangement, visualization, and features, the goal is to make it as simple as possible, and hopefully understandable by everyone through simple icons, textual, and shape visualizations, to encourage all residents to actively participate, through a coherent and consistent navigation and a user-friendly visualization and interaction layout.

For a successful use of a mobile application, by novice, elder, disabled, or to simply encourage the use by most of people is important to address usability issues. Design guidelines such as Web Content Accessibility Guidelines (WCAG) 2.1 [5] are embedded in numerous development frameworks (Ionic [6], Material.io [1] etc.).

Although, the graphical user interface (GUI) design follows material design for all BIMERR mobile applications, it is considered essential to note the main principles of universal design so that at any time

and at any point of design and development process, every step can be reviewed to avoid risks and inconsistencies.

The main principles of universal design are the following [4]:

Principle Name	Description
Equitable use	The design is appealing and useful to people of diverse abilities
Flexibility	The design supports individual preferences and ways of use
Simple and intuitive	The design is easy to understand regardless of user's knowledge, experience and cultural background, and current concentration level
Perceptible information	The design communicates the necessary information regardless situations, conditions, or user's abilities
Tolerance for error	The design minimizes hazards and the adverse consequences of accidental or unintended action
Low physical effort and need to remember	The design supports an efficient and easy use by the user, with a minimum of fatigue
Appropriate size and space use	All user should reach and manipulate successfully all the interaction points

For now, no specific usability issue will be addressed, other than those that are highlighted in the guidelines of material design [1], on which the structure and layout of the GUI is relied. Material design is a widely used framework, most users are familiar with from previous experiences of mobile applications usage, and is mainly used for mobile responsive applications, as it already integrates universal design conventions and usability guidelines addressing “all”. Any specific usability issue and its importance will be addressed in future versions of the application.

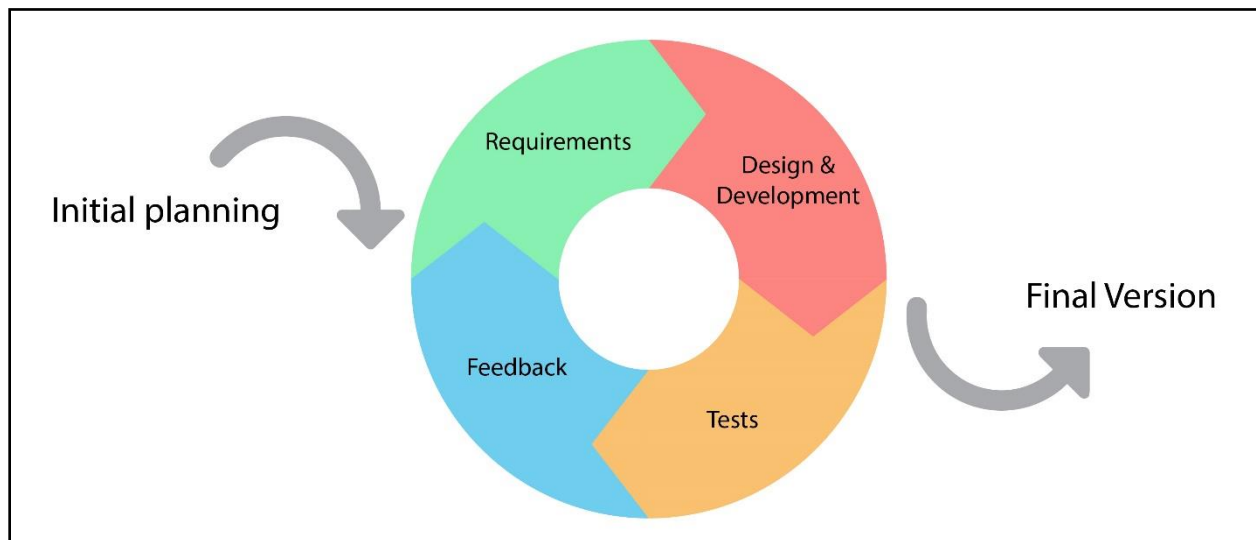
## 2.4. GUI DESIGN METHODOLOGY

There are many design and development process models (e.g. waterfall, V-model etc.) [7], or approaches that have been recommended throughout the years and can be applied in all or different processes (design, coding, testing etc.) of either design or developments that have been evolved and are used in different manners in various fields tasks (e.g. agile or other iterative incremental development models, universal design theory etc.). Some theoretical approaches are abstract and thus more flexible and open to adjustments, while others are stricter depending on the nature and goals of a procedure [8].

A generic iterative design process has been followed for two main reasons. First of all, as already mentioned above, users are “all”, and as such, there are countless ways they might use their space, and their reactions and participation to the equally dynamic nature of renovation workflow do not set up clear, linear steps and processes. Subsequently, an effort to propose specific design choice (visualization,

components, interactions, content correlation etc.) and “lock” them at an early stage would be risky and unnecessary. The second reason is that PWMA application is heavily depended on other modules and applications, ordering for a highly flexible design process that would allow constant flow of information and feedback among consortium partners. Nonetheless, not much is expected to change on the content and layout of the main screens that are designed and developed for this first iteration, as presented in the current deliverable, whereas, expanding the design in detail and screen-wise, is expected to affect those main screens up to a point [9].

Consequently, an iterative and incremental design and development approach is adapted as seen in Figure 2. Every iteration circle starts from re-defining requirements and ends with a visualization of revised mock-ups when necessary and a developed application version. In every new iteration the application will be expanded, as new screens and functionalities will be elaborated, and the input from end-users and the discussion among the consortium partners will be expanded in depth with more detailed and accurate content. Any posed restriction or priority can constantly be under review, allowing a dynamic design and gradual development as functionalities of BIMERR components will continue to evolve in a more detailed manner (Figure 3).



**Figure 2 Iterative Design & Development Methodology**

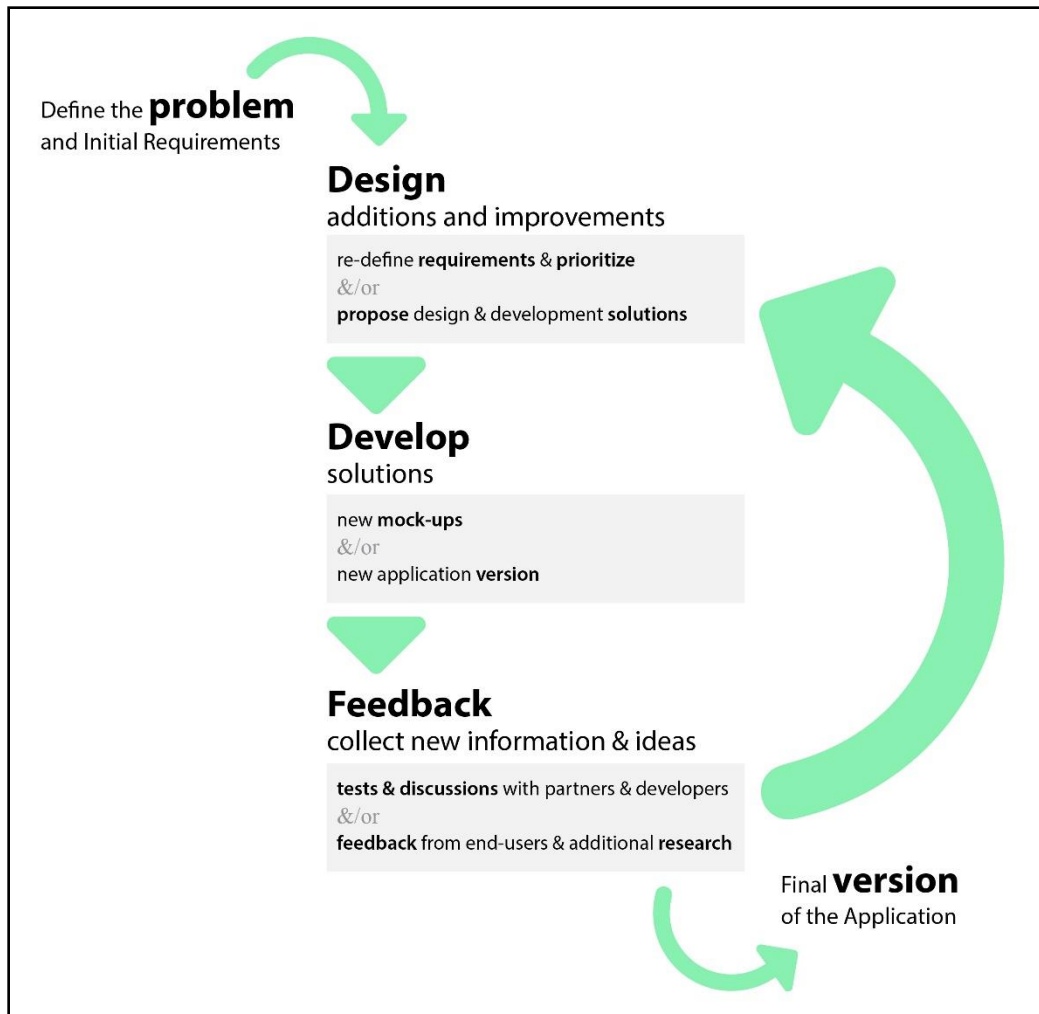


Figure 3 Iteration circle of Design & Development Methodology



### **3. USER REQUIREMENTS AND USE CASES**

#### **3.1. INTRODUCTION**

The first iteration is the content of this deliverable. The initial BIMERR Building Occupants requirements identification (deliverable 3.1), lead to the statement of the main goal and functions of the application (section 2.2) that were discussed among the consortium partners to define the first requirements of PWMA application for residents (section 3.2).

Based on those, as a first step – design, CERTH designers team used Personas [19] to frame the context of the application use and visualize the main expected screens and layout of the application. Designers and developers of CERTH further elaborated the content of Mock-ups and GUI components, which also lead to a first version of Information Architecture (Figure 5) and Navigation Diagram (Figure 6) (chapter 3). On the second step – development, gaps and inconsistencies were identified by the development team, who tackled minor issues with or without the consultation of designers. Other issues, questions and gaps that were identified and related either to back-end or front-end, are taken into account and will be prioritized for the next iterations based on feedback, discussion and further research among the partners and the end-users. That feedback and additional information collection is the last step of the iteration and the outcome will feed the next iterations (Figure 3).

At this point specifically, after the questions arose from the first will be discussed with the rest of the project team or will be part of the next phase that includes meetings with the residents. Ideally, during those meetings, some screenshots and the main workflow will be introduced to the end-users in the form of situated scenarios. Questionnaires will be used to check the validity of the scenarios and provide feedback on the proposed GUI features. The conditions and means will depend probably on remote communication or planned differently for each pilot and be managed locally. This is under discussion and will depend on local situations regarding COVID-19. The goal is to use the personas-based scenarios to gradually elaborate the interface main screens (based on hierarchy of functionalities importance), the features and a flow of use (navigation) [10], [11].

## **3.2. OCCUPANTS' REQUIREMENTS**

In the following sections, the main user groups, functional and non-functional requirements of the application, types of devices used and the way these correspond to design choices will be outlined.

### **3.2.1. Requirements**

The first requirements emerged from the use cases, the main goal and the three main functionalities of PWMA application for residents, as already described in section 2.2.

#### **Users**

The application for residents will be used by the user group “Building Occupant”. Secondly, it is possible that this application would also be used by someone with a similar role as a “Resident delegate”, and by the “Building Owner”. As resident or residents’ delegate, someone would have access to the exact functionalities as a resident simply after communicating with residents and using the app instead of them or summarizing the reports from many residents. The building owner has full authority to control the renovation project, monitors all costs, and tasks using other tools, but he/she should be able to use the application for residents as well.

#### **Device**

- The application for residents will be used via any android-based tablet/mobile phone.
- It will be responsive.

#### **Functional requirements**

- Online operation
- Photo/Video Capture
- Text Annotation
- Voice Annotation
- Notification System

#### **User requirements**

- The user should be able to view an overview of the tasks that are scheduled or on going in his/her building/location. Mainly be informed and notified regarding H&S concerns, the location, time and duration of all tasks. The overview should be given in a visual and user-friendly manner.
- The resident should receive notifications about an issue's status and about changes made during a task, related to H&S, location, time, duration.
- The resident should be able to reply to notifications on H&S concerns and Work Schedule that may affect residents' zone/apartment etc. These are forwarded to the foreman/project manager etc.
- The user should be able to take photographs, videos, text notes or voice notes regarding a renovation task and send them as a reported issue.
- The user should be able to create his/her own issue that comes up during a renovation work or an issue observed by occupants visiting the renovation site, such as dangers, malfunctions etc. (even while living in the renovation site after the renovation is completed).
- The user should be able to view the actual progress of the Works, and possible deviations, such as delays from the planned progress, and receive notifications accordingly.
- The user should be able to configure the type of notifications he/she would prefer to receive.
- Costs and financial aspects should be part of this interface only in case the residents do pay for the renovations.

### **3.2.2. Personas and use case stories**

Personas and Use case paths are described below to depict the users' needs and requirements. Application functionalities of the system and their relationships should match the needs of the personas and give solutions to these user needs. Those scenarios, may be slightly revised based on the upcoming meetings with the residents and may be adjusted to pilot specific needs. Nevertheless, the application is to solve these problems, thus they can be used as a validation point for the design and development of the application.

#### **Persona 1- Example Scenario**

**I want to:** Be always aware of any possible danger, inconvenience, health and safety matter prior to renovation task's start and during it

**The building:** Residence Building

**The user:** Parent of three, lives on the second floor

**The use or habit:** Drives and parks the family car in front of the entrance every morning at around 7.30. The parent puts the baby inside, and the other two young kids. He/she takes them to the kindergarten and school before going to work.

**The renovation task:** floor removal of floor on the third – debris removal from the balcony with the use of crane, at the street closed to the building entrance.

**The problem:** Danger for people crossing in and out of building. The parent happened to talk with some workers and was notified about next day's plans. The resident is concerned mainly about the kids and the baby, as he/she carries the baby to the baby seat, and the older kids walk around by themselves before getting into the car. Another concern is the damaging of the car. He/she will complain to the renovation company because he/she was not informed. Now, he/she loses valuable time because he/she has to park the car further and make sure that a neighbor will assist in overlooking the kids.

### **Context of application use**

The user needs to be informed for every scheduled renovation task, and the health and safety effects that are planned and expected during a task. This way, residents are able to plan for themselves and take actions prior to a task or be aware of important health and safety issues during one. For example, they might have to close doors or windows, move obstacles or personal items that could be damaged during renovation tasks, keep in mind that facilities will be out of order, keep themselves and their children safe etc.

### **Persona 2 - Example Scenario**

**I want to:** Be always aware of changes at any possible danger, inconvenience, health and safety matter prior to renovation task's start and during it

The building: Residence Building

The user: Single working resident, lives on the third floor

The use or habit: He/she works at 9.00 in the morning. As new at work usually spends up to 9 hours at work and other two for the total transportation. Giving much priority at work and healthy diet, it is an essential routine for him/her to cook every day starting at around 6.30 in the morning to have some cooked food once he/she enters the house late in the afternoon.

The renovation task: The electricity is due to be cut from 8.00 to 10.00 in the morning for the whole building. An announcement poster is hanged on the entrance five days ago.

The problem: The residents is upset because while cooking at 7.45 the power was cut. No announcement about any change was made by the renovation company, because it was an on-the-spot decision to cut the power 15 minutes earlier. The workers notified only some of the residents they met this morning in and out of the building.

**Context of application use:** The user needs to be informed on task progress, the deviation between actual and planned dates/times, and any change that the manager could make along the way (e.g. add a new effect or change the scheduled date/time), once the planning process is a dynamic process. Re-planning practices that are possible based on feedback from workers and residents and unexpected events, should be at all times evident to everyone that uses the spaces under renovation. More specifically, receiving a notification once a new task is scheduled (pending status), some minutes before it actually starts (on going status) and when it is actually completed (completed status) is of major importance so the residents can plan for themselves “on the fly” and be as flexible as possible in their daily routine and use of space.

### **Persona 3 - Example Scenario**

**I want to:** Reach the workers and renovation managers if needed

The building: Residence Building, lives on the fourth floor

The user: The residents’ delegate

The use or habit: One of the stuffs of the cleaning crew, notified the residents’ delegate that one window is not working properly.

The renovation task: New windows where installed in the shared space of the building stairs one week ago. The renovation is completed and no worker or site manager is around.

The problem: The residents' delegate communicates with the renovation company and tries to explain which one is the window that does not work properly and why. They together spend significant time expanded in three-day period to decide whether the problem was a bad installation or a misuse from the cleaning stuff. A person from the renovation company comes over and takes a photo from the window, which he/she sends to the manager. The manager finally decides that this is indeed a bad installation issue, which they now need to tackle in the near future.

### **Context of application use**

The user needs to be able to reply to the manager by filling a comment form (type text, add photo/video, audio message) regarding complaints, comments, inform about observations that he/she makes while using the space. Those are taking into account by the manager to make planning changes if required and keep an open dialog between renovation managers and residents. Residents, this way, contribute to the dynamic process of renovation planning.

### **Persona 4 - Example Scenario**

**I want to:** My opinion, experience and story matters

The building: Residence Building

The user: Young resident – student, lives on the first floor

The use or habit: He/she uses the bathroom shower a lot. He/she also uses a dishwasher machine.

The renovation task: Plumping renovation in the apartments of the first floor completed about two weeks ago. There are three apartments on the floor. The other two are currently unoccupied.

The problem: The resident noticed that although hot and cold water in the bathroom taps have equally strong water flow, this is not the case for the kitchen tap. The cold water has significantly lower flow than the hot water. Although he/she is not sure if it has to do with the dishwasher machine. He/she does not know if this was supposed to be or not, and no one lives in any of the other apartments that could have the same issue in order to ask or check it out. He/she does not know if should refer to a plumber or to the renovation company.

### **Context of application use**

**Deliverable D6.10 ■ 08/2020 ■ CERTH**

Page 30 of 71

**BIMERR project ■ GA #820621**

The user needs to be able to confirm or comment on the completion of a task in a similar way, as a feedback to the manager. No task has actually finished if not proven to have led to a fully functional component, system or service. Workers and managers themselves proceed to checks and test on completion of a task. However, it is a common truth that a design and building process is not fully evaluated, unless the people that actually use the space can observe the outcome and comment on the quality, sometimes even for a long period after the completion.

### **Persona 5 - Example Scenario**

**I want to:** Be actively involved in planning process of improvement of the spaces I use

The building: Residence Building

The user: Residents/Owner, lives on the third floor

The use or habit: Always uses the stairs as a work out

The renovation task: The building is under renovation. Shared spaces on the roof and some apartments will be partially renovated in areas like balconies, focusing on the outside of the building. The inner stairwell is not planned to be affected in any way.

The problem: The resident/owner runs up the stairs holding the head up. He/she trips on the last step of a stair, as it is slightly of a different height than the rest. He/she did not hurt significantly, as it happens very often. He/she approach the workers and proposes to use the opportunity of renovation to do something about the stairs. He/she has to also talk with the other residents and owners to reach an agreement towards an intervention at the stairwell that might be cheaper and less time consuming if done now, thinking that they may have also suggestions for additional issues to tackle. They all seem preoccupied. Maybe he/she will end-up talking and negotiating with the renovation company first before he/she communicates back with the residents/owners of the building.

### **Context of application use**

The user needs to be able to send an observation to the renovation manager, annotating an issue that is not necessarily related to a task that has been scheduled.

### 3.2.3. Information Architecture & Navigation

The following section will provide a usage scenario of the application that correspond to UI workflow and navigation methodologies used to cater for those scenarios and explain the decisions made by the design team regarding Information Architecture and Navigation.

**Usage Scenario:** An apartment resident would like to be informed about planned health & safety issues and annotate issues, but a building owner/occupant would like to also be informed about the completion progress of a task and whether its start or finish time is delayed or was on time.

Consequently, the main entity is a task of a project once the time, the location and the H&S issues related to it is the important information that change when a task changes. The home page is an overview of all the tasks (pending, on going, completed) in the premises of the user. The user can see more details my tapping on one of the tasks. The next important functionalities are those of reporting new issues, viewing and following incoming notifications, and finally the settings. All main actions are directly accessible from tabs at the lower part of the screen, and all secondary are placed in the side navigation menu at the top left of the screen (fig. 4).

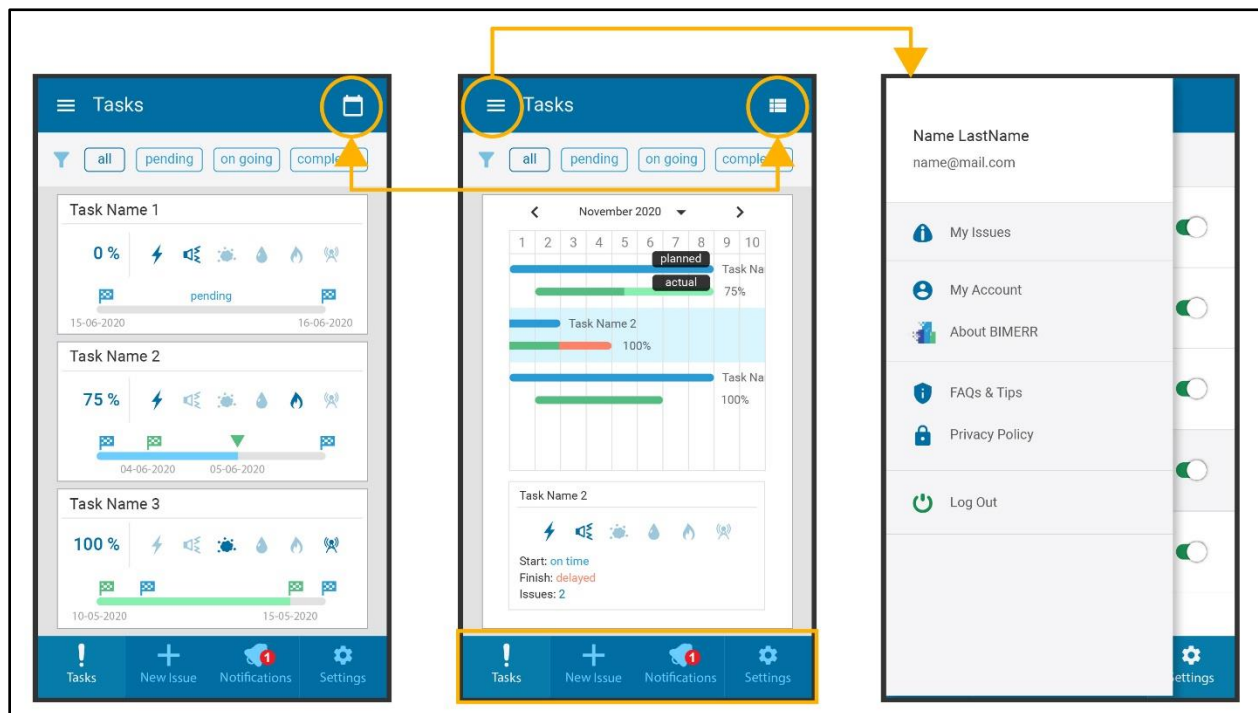


Figure 4 Home page “Tasks”- List view (left), Calendar view (center), and Menu (Navigation Drawer) (right)



As seen in the Information Architecture (fig.5), there are four main tabs (grey rectangles): “Tasks” – screens indicated with numbers 3 and 4 in the navigation diagram (fig.6), “New Issue” - screens 6,7,8, “Notifications” – screen 9, “Settings” – screen 10, all reachable from the lower menu as seen in figure 4. Those lower tabs buttons and the menu button (navigation drawer) are always visible and directly accessible in the first navigation level.

The home page “Tasks”, is an overview, displaying a summary (status, H&S effects, date and time, locations etc.) of all pending, ongoing, completed renovation tasks in his/her building. The user can view the tasks in a form of list in default “List view”, or change view in “Calendar view” where tasks are displayed in a way that the user can see them all across a month, a week, or a day, overlapping times and times between them (fig.4). In both views, the user can use the filter to view on of the tasks’ categories: all, pending, on going, completed.

From the “Tasks” tab the user can select one task to enter and view its details into a second navigation level in “Task Details”. A “back” button is needed to go back to the first navigation level the home page “Tasks”, either it was in “List” or “Calendar” view mode (fig.6, screen no.5).

From the “New Issue” tab the user can fill a form to submit an issue that he/she has observed regardless of its relation to a scheduled task. The user follows three successive steps to fill the report form: 1 – confirm location ((fig.6, screen no.6), 2 – select type of issue (fig.6, screen no.7), 3 – adds description including text, video, photos and/or voice message (fig.6, screen no.8), and taps the “submit” button to send the report.

From the list menu of the navigation drawer the user can enter the list of all the issues that he/she has submitted from “My Issues”. By selecting one he/she can enter the third navigation level and view the “Issue Details” in a modal window. An “x” button is needed to close the window.

From the “Notifications” tab the user can view a list of all the notifications that he/she has received. It is possible to delete them too. Every notification leads to the “Task Details” page of the task the notification is referring to.

From the “Settings” tab the user can enable/disable different types of notifications that he/she would like to receive. Even if notifications are disabled the user can, at any time, be informed from the home page of “Tasks” and view every “Task Details”, as those pages are refreshed in real time.

From the list menu of the navigation drawer the user can enter secondary pages and/or seldom used functions. By selecting “My account” from the side menu, the user can enter his/her user account (mail, name, building etc.), be linked to the website of BIMERR project through “About BIMERR”, enter the page

“FAQs & Tips”, or the “Privacy Policy” page. A “back” button in those pages is needed to go back to the first navigation level.

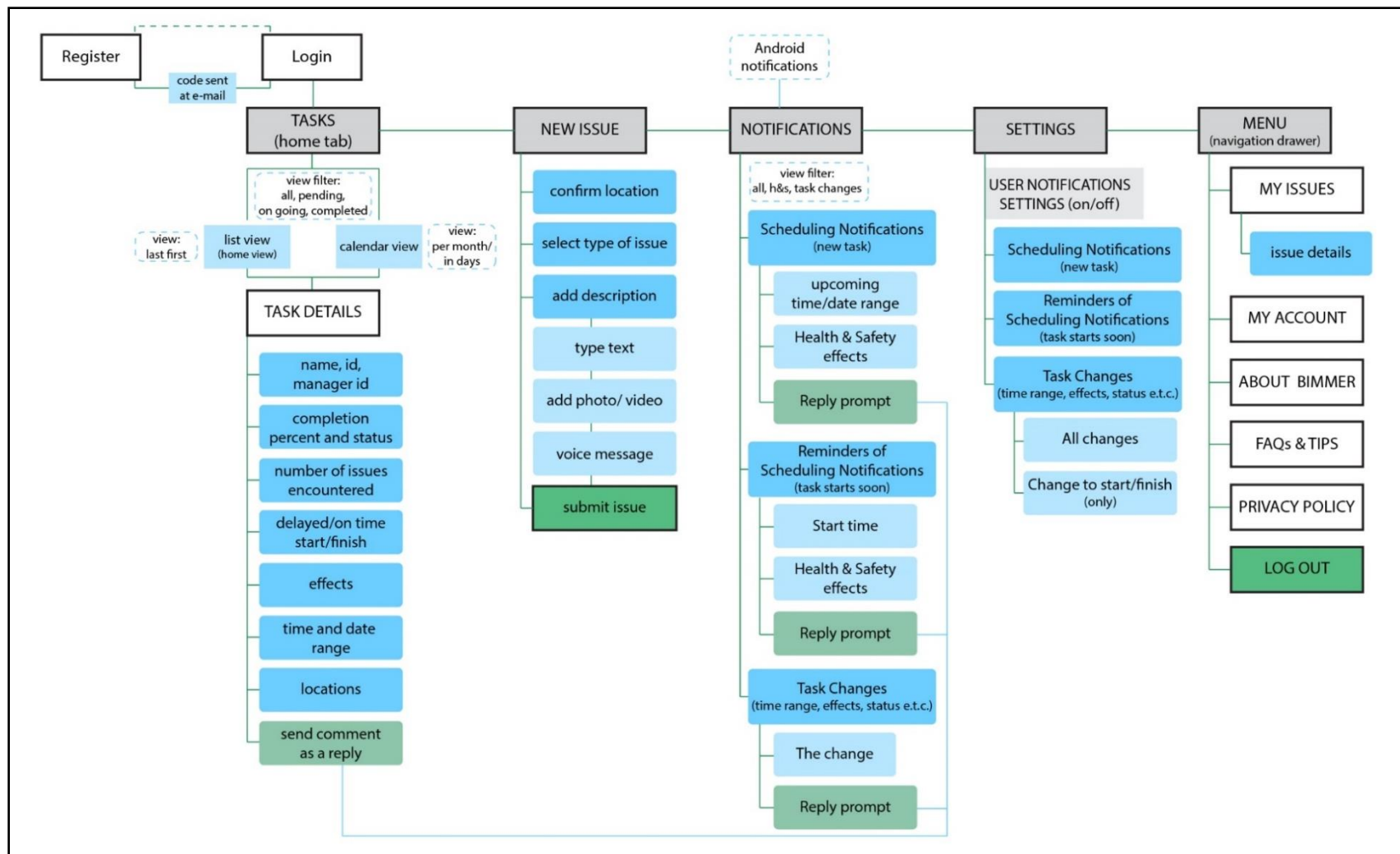


Figure 5 Information Architecture



Figure 6 Navigation Diagram – Mock-up screens

### 3.2.4. Design Choices - Layout

We tried for simple and consistent navigation, layout, information visualization and content configuration. Regarding visualization we mainly focused on two general aspects: a) Color contrast for text against background, b) Variation of text, icon and color combinations. For doing so, we also consulted some aspects of WCAG 2.1 guidelines [5] and the respective criteria. All icons, texts and even layout will and should be redefined as the content and functionalities continue to evolve along the design and development process.

- A flat design was implemented, and visual components (e.g. highlighted icons, progress bars) were used to summarize in a little space all the important information. A combination of text, descriptive icon and indicative colors, was used whenever was necessary and sufficient. For example, to summarize all the information related to a task in the overview “Task” page, it was essential to use descriptive icons and visualizations due to small mobile screen. Every type of effect is displayed with a descriptive icon. If a task affects one or more areas such as power cut, noise, dust, water cut, gas cut, communications cut, then the respective icons are displayed at 100% opacity level. The areas that are not affected are displayed with 36% opacity.

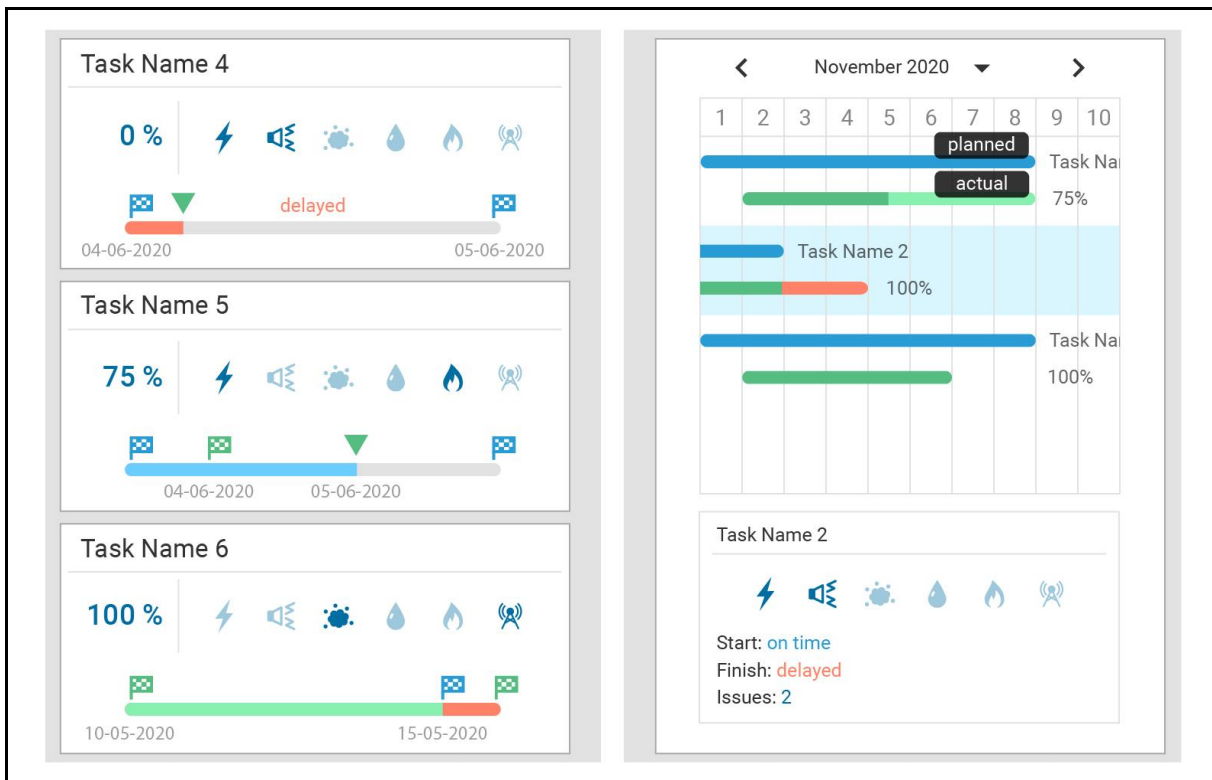
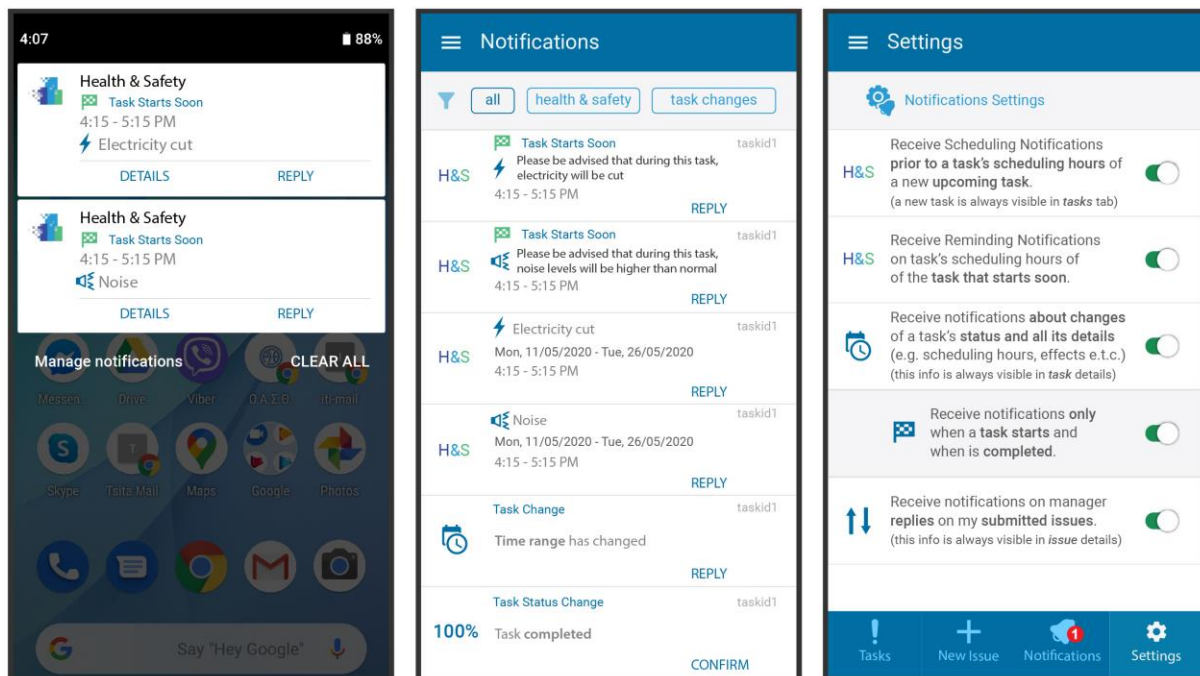


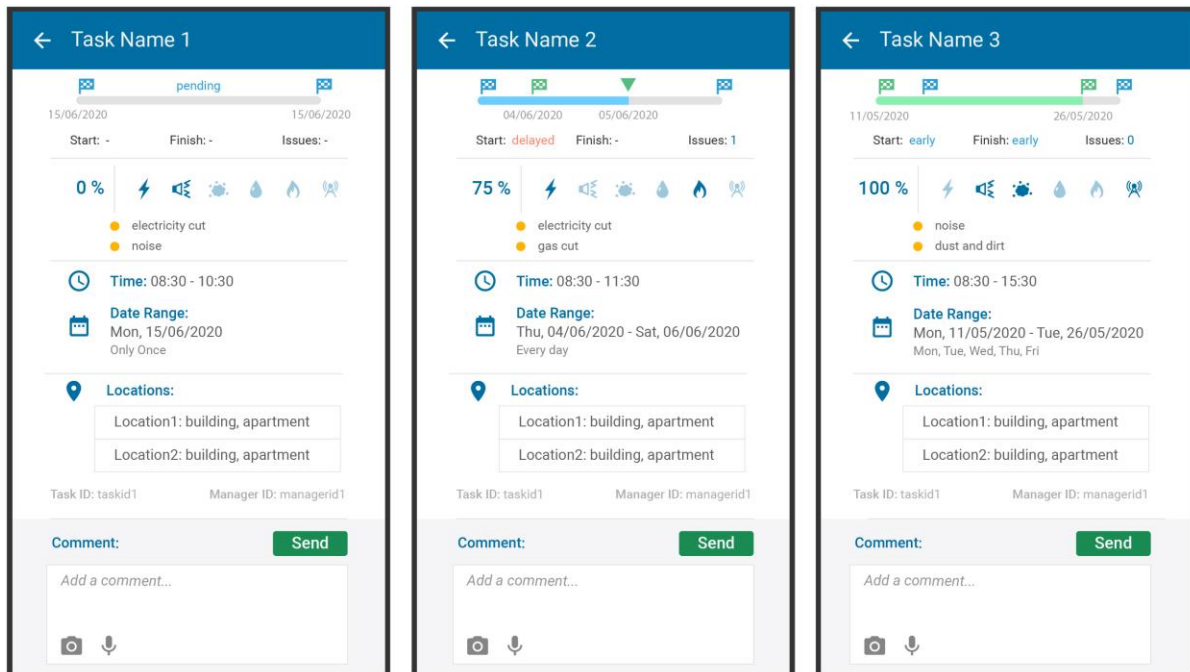
Figure 7 Mock-ups, Colors and Indicators Home page “Tasks” - List view (left), Calendar view (right)

- The completion percentage is displayed both with numerical indicator and in a completion bar. On the completion bar which indicates the planned time duration, blue flags indicate the planned start and end of a task's duration, and green flags indicate the actual start and actual end of a task. A task that has not yet started is indicated with light grey, one that is ongoing with blue and a completed task with green. Deviations in actual start and finish time from the planned are indicated with red.
- Two different view modes are suggested for the main overview screen (list view and calendar view as a simple Gantt chart with progress bars) (fig.4). In the calendar view, two bars are used to convey information about one task: a bar of the planned time and a bar for the actual time. In this view, the user can visually relate tasks with each other regarding their times. The user with "finger tap" and "scroll" left and right can view all the weeks of selected month at a day level. By selecting a day the user "zooms" to view a day at the level of hours. List view facilitates a continuous "scroll down" for a fast overview, while calendar view focuses on interrelating tasks with each other. A task in the calendar view is a clickable "item" that lead to a page that displays more details on the selected task. Similarly, the user enters a task's details by tapping on a list item of the tasks' list view mode.
- Direct access to any last change is essential for the user. Android notifications will be used. By clicking on the notification, the user enters the details of the task the notification refers to, either it is a new or an older task. Keeping track of tasks and changes, through a linear logging of actions is important. The notification tab, displays all received notifications, read or unread, in time order supporting the user to follow up. Text-based components and small icons are used to convey messages for notification alerts, while keeping the language simple. The user will also be able to manage his/her notifications and turn them on/off, a functionality that allows customization according to user's preferences from the "Settings" page and the Notifications Setting). (fig.8).



**Figure 8 Mock-ups, Android Notifications (left), “Notifications” page (center), “Settings”- Notifications Settings (right)**

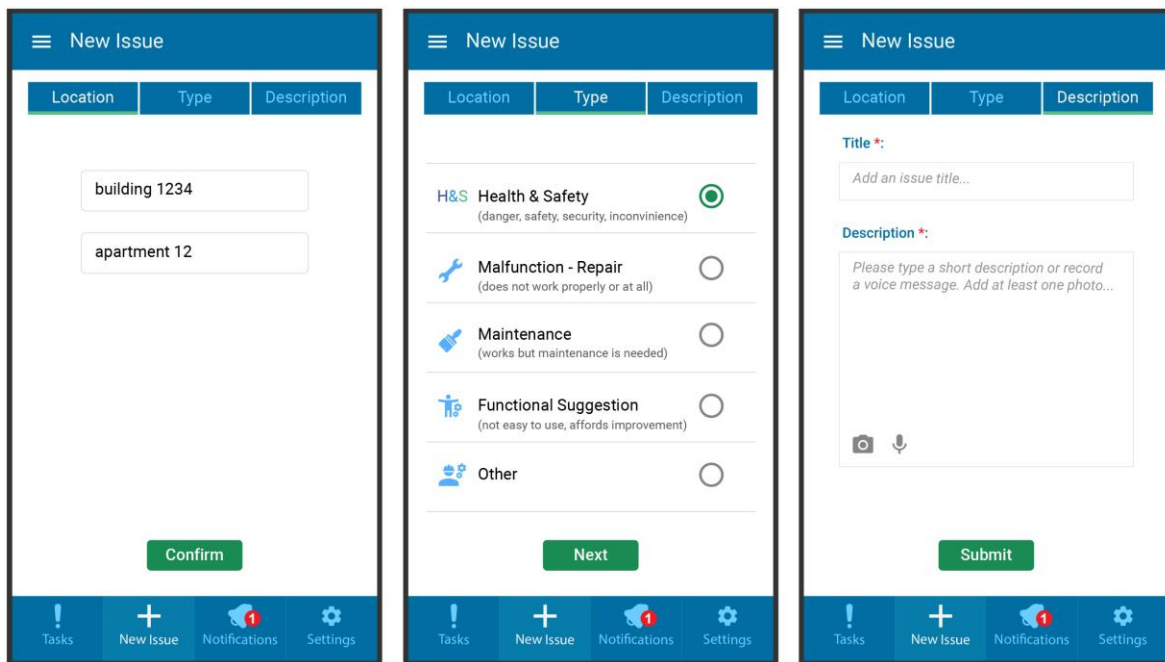
- In “Task Details”, some of task information is task name, completion percent (%), health & safety affects (icons and textual), completion status (pending, ongoing, completed), planned/actual deviation etc. Although, not all those are yet defined, the idea is to prioritize as the following statement indicates: A task at “this” status has “these” effects, during “this” time/day, at “these” specific locations. The user is encouraged in the same page to directly reply and send a comment: type text, add photo/video, voice message.



**Figure 9 Mock-Ups, Selected task – Details of tasks (second navigation level)**

- The user can report any issue selecting one of the following categories, such as Health & Safety, malfunction, maintenance, functional suggestion. The user adds a title and a description that consists of text or voice message, and it is necessary to add at least one photo. It is possible to add more than one photo or a video. He/she then submits the issue (fig.10).





**Figure 10 Mock-Ups, “New Issue” page: Location confirmation (Left), Type of issue selection (Center), Add Description (Right)**

### 3.3. USE CASES

Based on the personas scenarios, and the layout that is presented above, the main flow of users interactions with PWMA application lead to the use case scenarios presented in this chapter. The main functionalities are the notification system, the responses of the users related to specific tasks in form of comments (feedback, concerns etc.), and the report of any observed issue in the user’s premises. The main steps of the users’ interactions through the GUI are described in three use case scenarios.

#### 3.3.1. UC – 01: Notifications and residents’ responses related to tasks

**Related Persona(s):** 1 - Be always aware of any possible danger, inconvenience, health and safety matter prior to renovation task’s start and during it, 2 - Be always aware of changes at any possible danger, inconvenience, health and safety matter prior to renovation task’s start and during it

1. Once the manager publishes a new renovation task, or publishes a change of an existed renovation task's parameters (e.g. scheduled time) the systems filters the users that are registered under the premises where the renovation will take place.
2. The system filters the notification settings of the users and sends the appropriate type of notifications according to their notification settings.
3. The user according to his/her notifications settings receives a notification (android push notification) to his/her device.
4. The user enters the application by clicking on the received notification about the newly scheduled upcoming task, or for a new change on a previously scheduled one, and views its details: the effect (H&S issue, caution), the planned date and time of start and the end of the task, the specific area of the premises. The user can act and tackle any schedule arrangements that affect his/her routine accordingly, prior to the task and avoid any further inconvenience of the last minute.

### **3.3.2. UC – 02: Residents' responses related to a specific task**

**Related Persona(s):** 3 - Reach the workers and renovation managers if needed, 4 - My opinion, experience and story matters

1. The user notices an issue in his/her premises. He/she checks it and realises that it is related to a renovation task that has been completed in the past. The user can find the related task in his/her "Tasks" overview page, and select that task to view its details.
2. The user can also find the completion task from the notification he/she had received on the completion of the task, in his "Notifications" page, and select it to view the task's details.
3. The user enters the task's page. He/she can add a comment (text, photo, video, voice message) and send it to the manager as a feedback report based on his observation as related respond to the task completion.

### **3.3.3. UC – 03: Residents' new issue reports**

**Related Persona(s):** 5 - Be actively involved in planning process of improvement of the spaces I use

1. The user notices an issue in his/her premises.
2. The user can report a new issue through the “New Issue” page, where he/she fills the form (location, type of issues, description – text, photo, video, voice message) and send it to the renovation manager.
3. The manager can examine the new issue report and act towards scheduling it for the future or include it in an on going or a pending task.

## **4. DEVELOPMENT PLAN AND TECHNICAL REQUIREMENTS**

This chapter contains a view of the methodology followed for the development plan and the technical requirements, due to the fact that after the design process and the consideration of use cases and user requirements, the aforementioned procedures take place.

### **4.1 DEVELOPMENT PLAN**

Based on the scope, timeframe, complexity level and overall purpose of the PWMA Residents' Application, the development team decided to follow a development methodology which:

- Welcomes changing requirements, even in late development.
- Delivers working software frequently (weeks rather than months)
- Working software is the primary measure of progress
- Supports sustainable development, able to maintain a constant pace
- Pays continuous attention to technical excellence and good design
- Has simplicity—the art of maximizing the amount of work not done—is essential
- The best architectures, requirements, and designs emerge from self-organizing teams
- On a regular basis, the team reflects on how to become more effective, and adjusts accordingly

These are some of the Manifesto for Agile Software Development principles, so the decision was to follow the Agile methodology. [12]

Initially, the mock - ups of design process and the user requirements and the use cases were examined. The conclusion of this examination was that there were three important sprints (steps) to be done. Sprints are iterative cycles where the functionality is developed new increments. Sprints are indispensable in agile methodologies, and getting sprints right will help the agile team ship better software with fewer caveats. These three sprints (steps), depend on the degree of difficulty of implementation and can be summarized as follows:

- The first sprint included the general set up of the user interface and the easier to be built screens.
- The second included functions dependent on back-end integration and deployment that requires other BIMERR modules to be in place in order for the app to function as designed
- The third contains also similar functions as mentioned before in the second sprint.

## **4.2 DEVELOPMENT TOOLSET**

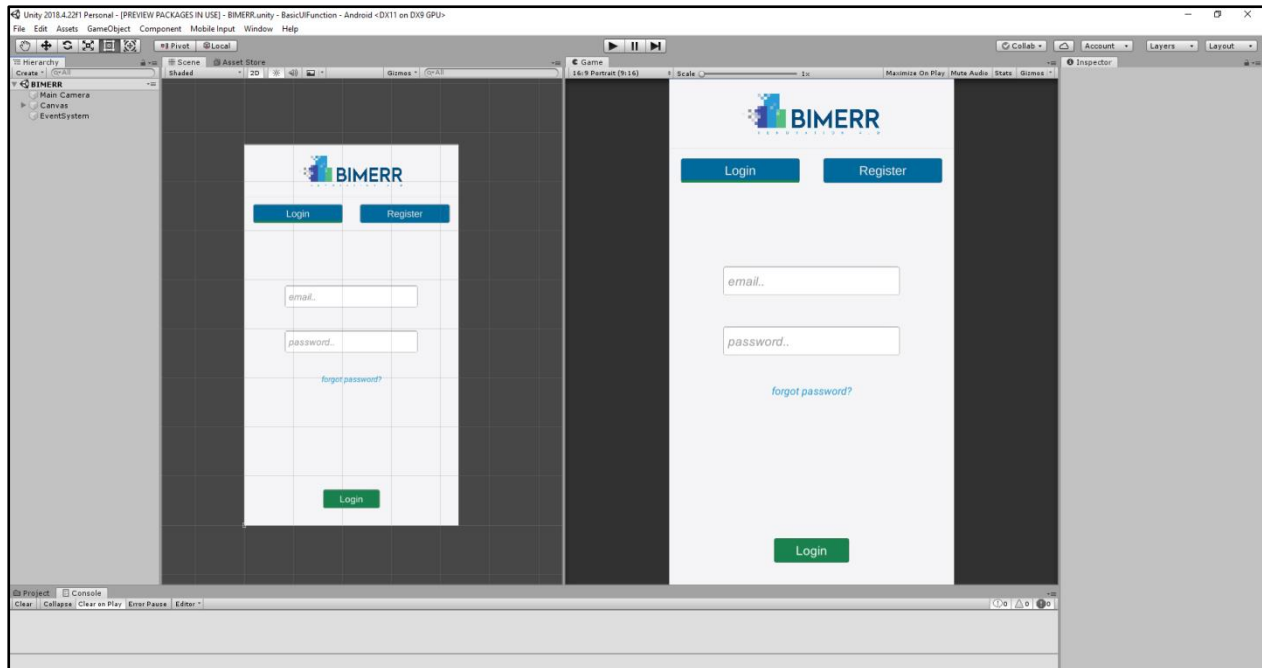
In this subchapter the development toolset and the software used during the application's development are described.

### **4.2.1 Unity**

The Unity platform was chosen for the building procedure. It is worthwhile to note that Unity was also used to develop the application "ARIBFA" (Augmented Reality enabled In-situ Building Feature Annotation) which also led by CERTH, because of the AR SDKs it supports. For better cooperation and compatibility between the two applications, it was important to use the same platform. Thus, Unity 2018.4.22 was the version that was chosen for App for residents, which is the same version that was used in ARIBFA development.

Unity is a Game Engine created by Unity Technologies in the C++ programming language. The first version came out in 2005. Its purpose was to make creating games easier and this area to be addressed to more developers. Although it was created to create games, it can also be used for various other types of applications such as PWMA App for residents. The engine can be used to build 3D and 2D applications, simulations and games as well as Virtual Reality (VR) and Augmented Reality (AR) applications such as ARIBFA app.

At the technical level, the Unity platform offers the possibility to its user, through its Editor, to create scenes, objects and interactions. It combines their design with the ability to program them (via the C# programming language, earlier versions also offered the possibility for programming in JavaScript) through scripts that are loaded into them. Also other functionalities offered by the platform are external plugins and drag and drop manipulations. As for the mobile games, since 2018 Unity had been used to create approximately half of the mobile games on the market and 60 percent of augmented reality and virtual reality content, including the emerging augmented reality platforms, such as Microsoft HoloLens [13], [14].



**Figure 11 Unity environment**

## 2.2.2 C#

As mentioned above Unity offers the object-oriented programming language C#. C# is a high-level, object-oriented programming language that is also built as an extension of C. This language has interoperability and rich library, it is scalable and updatable. Its code is easily portable and is fairly fast computing (certainly not at C / C++ levels) [15].

## 2.2.3 Visual studio

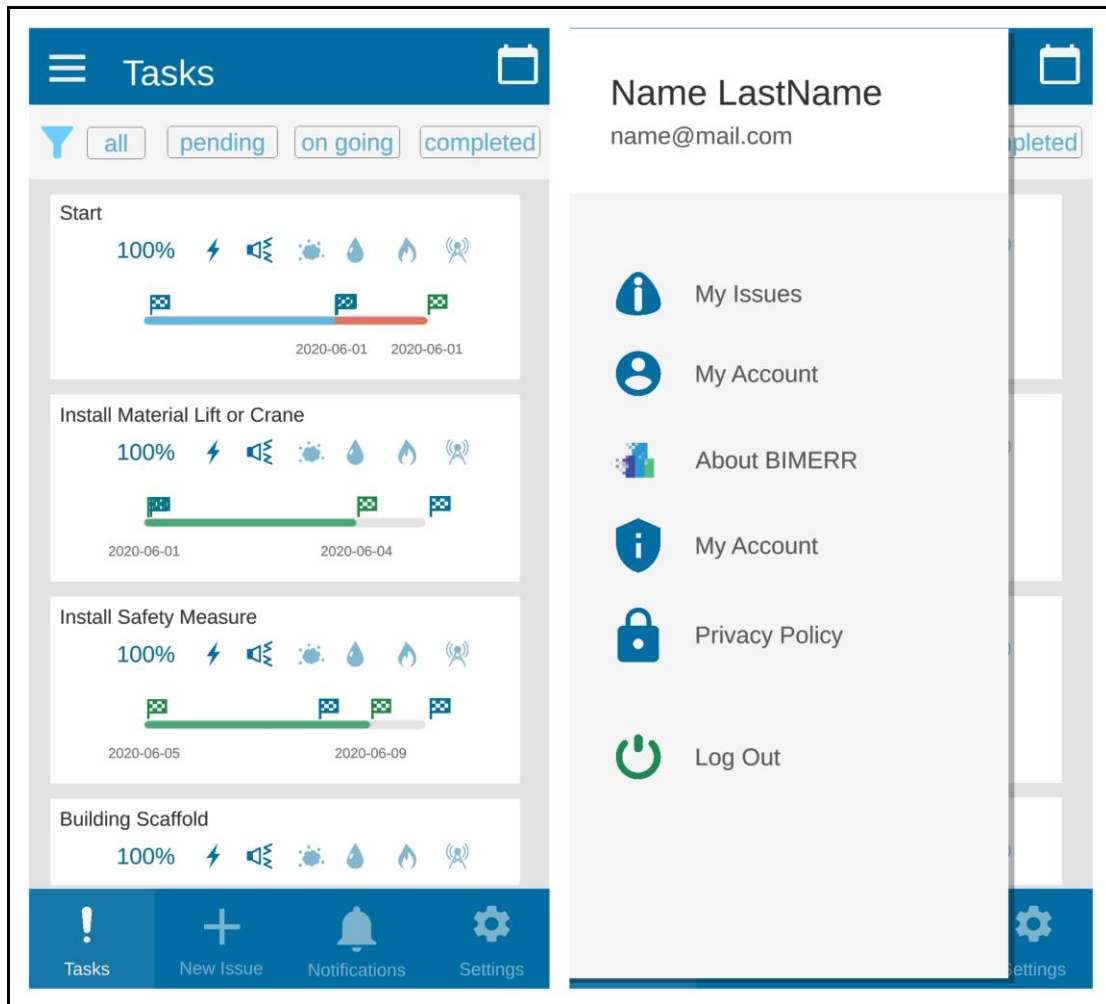
Visual Studio is a complete application development environment (IDE) developed by Microsoft. It is the recommended IDE for code development by Unity and they work extremely well together. Its Editor supports smart code completion techniques and code refactoring. It supports almost all programming languages, including C#, which is used to create scripts in Unity [16].

### **3 IMPLEMENTATION, FEATURES AND APPLICATION WORKFLOW**

In this chapter the implementation process, which comes after the development plan, is analyzed. Additionally, the features of the application are presented. Finally the application workflow is described in detail.

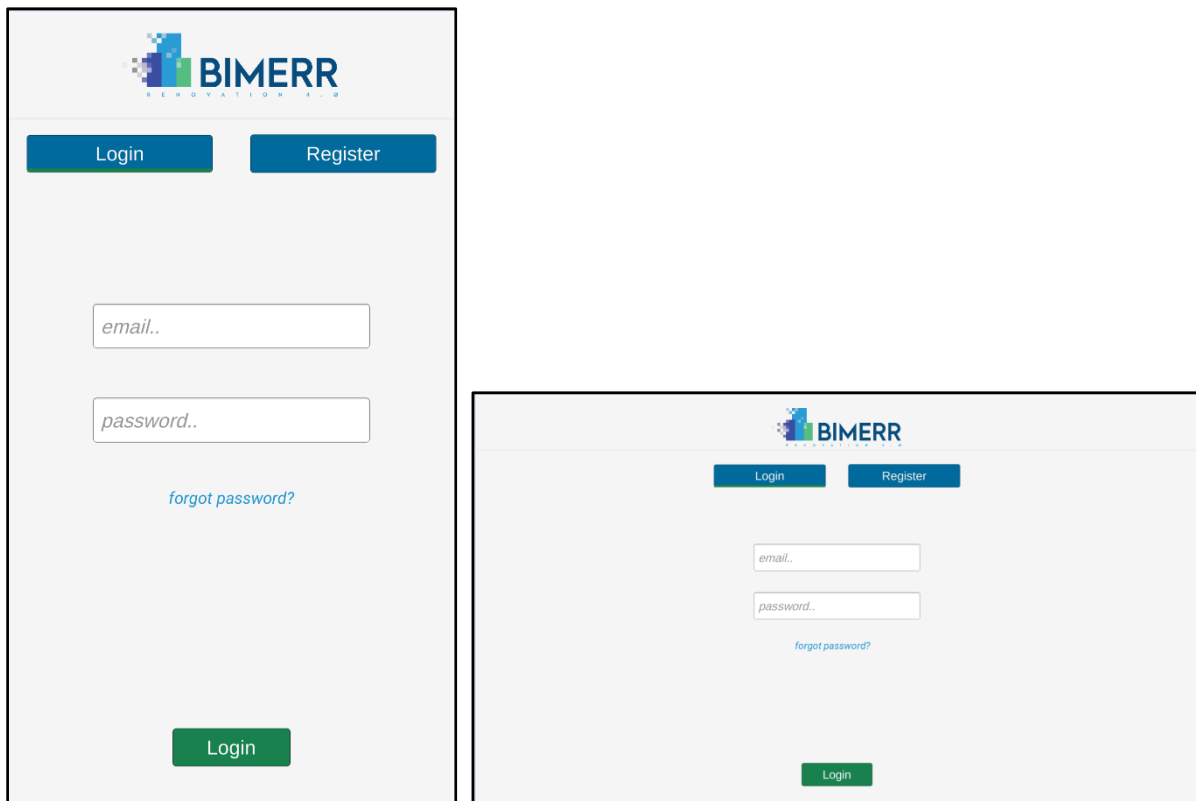
#### **5.1 IMPLEMENTATION**

After the development plan was laid out, the implementation of the application begun. The first step was to execute the first sprint. After reviewing the mock ups and other data, such as exact colors of images and buttons, exact size of fonts etc., the implementation of the screens begun. For this purpose Unity UI toolkit was used. Unity UI includes features like Canvas, interaction components, basic layout, animation integration etc. From this toolkit the Canvas which is the “parent” that holds all screens inside was used. From each screen an empty object (Game Object) was created, which acts a container for Components. Each game object contains interaction components such as buttons, sliders, input fields, scroll rects etc. A Button has an OnClick UnityEvent to define what it will do when clicked. For example, if the user is on the task menu and taps the “burger” button, the main menu will appear. In order to achieve this, an OnClick UnityEvent is defined that triggers the main menu game object (figure 5.1). All the remaining components that were needed, were created by taking advantage of Unity UI features. Further steps were taken in order to move UI elements at the correct position, depending on whether the device is on landscape or portrait mode. The result is a fully responsive GUI. One of Unity UI features is the Basic Layout, which contains Rect Transform. Rect Transform is a component that is used for all UI elements that has the following properties: position, rotation, scale and also width and height. With the right settings for each element in the Rect Transform and in the canvas scaler component of Canvas element, the GUI remains relative in every aspect ratio (figure 5.1.2) [17].



**Figure 12 Example of triggering the Main Menu**





**Figure 13** Example of the GUI which remains relative in every aspect ratio

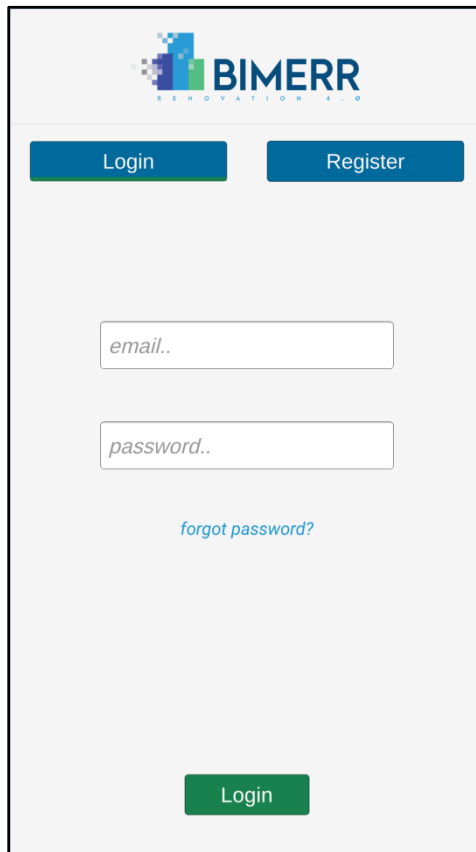
The data of the input fields has to be stored either locally or remotely, in order to be used. To achieve this, the data is converted to JSON (JavaScript Object Notation) file format. Correspondingly, the data related to user, such as task details are received in JSON file format and parsed. JSON is a readable format for structuring data. The parsing and the generating of them is easy to be executed by the machines [18].

## 5.2 FEATURES

All features have already been specified but some of them will be implemented in the second version, when it will be fully integrated with back end and all the other modules of BIMMER.

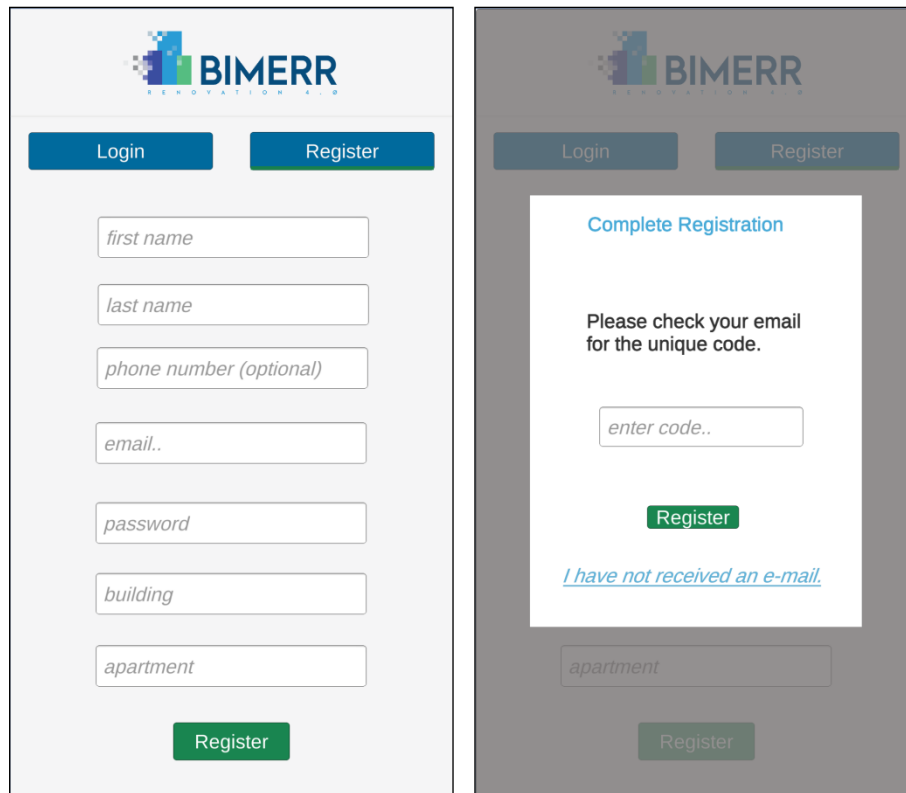
The first screen is the login window, which asks the user to log in with his/her assigned username and password. Initially, the user has to navigate to the register screen. Once the identifier is entered, the application stores and preloads it until it is re-installed or the configuration file is deleted.

After successful authentication, the user is logged in to the application and a set of permissions to access the application's functionalities is granted to the user.

The image shows a login screen for the BIMERR application. At the top, there is a logo consisting of a stylized 'B' made of blue and green squares, followed by the text 'BIMERR' in a bold, blue, sans-serif font. Below the logo, there are two buttons: 'Login' and 'Register', both in a blue gradient. In the center, there are two input fields: the first is labeled 'email..' and the second is labeled 'password..'. Below the password field, there is a link that says 'forgot password?'. At the bottom, there is a green 'Login' button.

**Figure 14 Login screen**

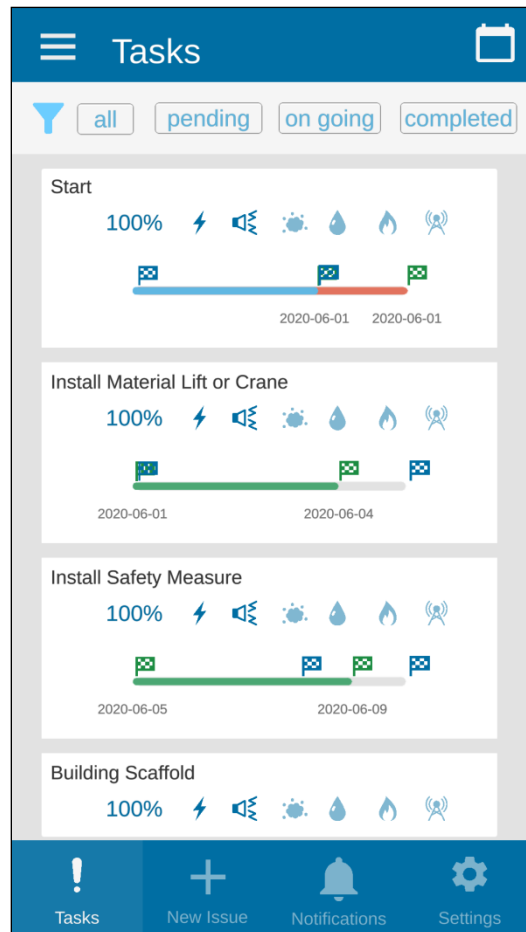
First time enrollment requires confirmation from the building manager. The user upon his registration is prompted to check his/her email for confirmation, which will include a unique code. The user will complete the enrollment by entering the code he/she received in the pop up window which will be prompted.



**Figure 15 Register screen (left), pop up window for finalizing registration (right)**

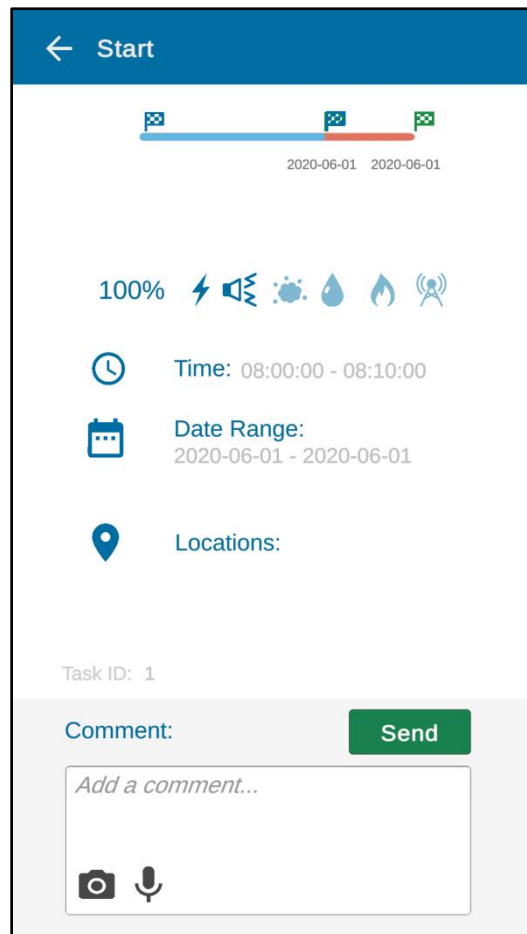
After the successful authentication, the list of tasks assigned to the logged in user is displayed. The list contains all the list items, ordered by the expected date and time of execution. The tasks that show up in the list are relevant to the building where the user lives.

User can see details of the assigned tasks, about planned and actual execution time and the completion percentage of them. In addition, he/she can see the health and safety issues that may arise during the execution of each task. Furthermore, user can see specific tasks that are pending, ongoing or completed according to what view filter is chosen.



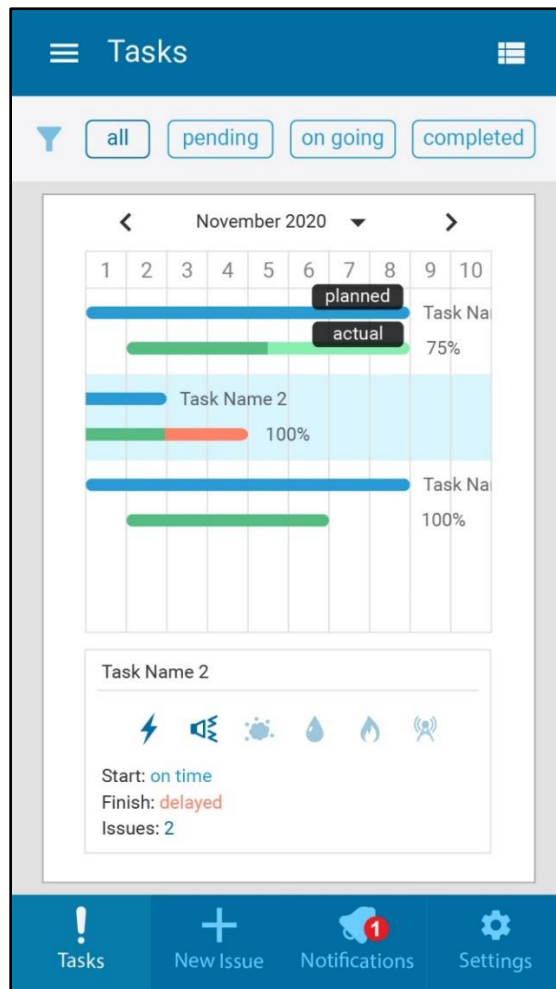
**Figure 16 Task tab – list view**

From the Tasks Tab the user can select one task to enter and view its details into a second navigation level in “Task Details”. In addition to the details which are displayed in List view, user can also see information about task id, task manager’s id, specific time range, locations affected. In addition, he/she can send a comment, which can include text, photo/video or even voice message.



**Figure 17 Selected task – Task Details, into the second navigation level**

From the top right corner the user can switch view mode between “List View” and “Calendar View”. The default view is “List View”. When user switches them, is navigated to a new window that he/she is able to view the tasks order by month and days. The user can check the details of a task by selecting the specific task from the calendar view list. The user will be able to see similar information with “Task Details” screen. The figure below presents the mock- up of the “Calendar View” and not a real screenshot of the application because the implementation of this window is not completed.



**Figure 18 Tasks Tab – Calendar View**

Tapping on the New Issue tab, user is navigated to the New Issue screen. He will be able to report (submit) issues to the AEC project manager or the AEC building maintenance manager, regardless of a scheduled task. He will also be able to submit potential issues he encounters in the building and make suggestions by completing a form with photographs, videos and text or voice description, at any time.

To fill the form the user must complete the following steps:

Step 1 – Confirm Location: The apartment and building the user is referring to, is already known from his/her account.

The screenshot shows a mobile application interface for creating a new issue. At the top is a blue header with a hamburger menu icon and the text 'New Issue'. Below the header is a table with three columns: 'Location', 'Type', and 'Description'. The 'Location' column contains two input fields: 'Building' with the value 'A' and 'Apartment' with the value '2'. A green 'Confirm' button is located at the bottom of the form.

Location	Type	Description
Building		
Apartment		

Confirm

**Figure 19 “New Issue” tab: Location confirmation**

Step 2 – Choose a Category: The user chooses a type of issue. This encourages the user to better define the issue and facilitates the filtering process of the receiving issues from the managers.

Issues are categorized based on the importance and the emergency for intervention into four types and are hierarchically presented to the user from the most important issue that needs immediate intervention, to the most secondary issue category. “Other” is added as an additional type for unforeseen issues.

- Health & Safety (danger, safety, security, inconvenience)
- Malfunction – Repair (does not work properly or at all)
- Maintenance (works but maintenance is needed)
- Functional Suggestion (not easy to use, affords improvement)

- Other

The screenshot shows a mobile application interface for creating a new issue. At the top is a blue header with a hamburger menu icon and the text 'New Issue'. Below the header is a table with three columns: 'Location', 'Type', and 'Description'. The 'Type' column is highlighted with a green underline. The table lists five issue types, each with an icon, a title, a subtitle, and a radio button:

Location	Type	Description
H&S	Health & Safety (danger, safety, security, inconvenience)	<input type="radio"/>
Wrench icon	Malfunction - Repair (does not work properly or at all)	<input type="radio"/>
Paintbrush icon	Maintenance (works but maintenance is needed)	<input type="radio"/>
Person with gear icon	Functional Suggestion (not easy to use, affords improvements)	<input checked="" type="radio"/>
Person with gear icon	Other	<input type="radio"/>

Below the table is a green 'Next' button. At the bottom of the screen is a blue navigation bar with four icons and labels: 'Tasks' (exclamation mark), 'New Issue' (plus sign), 'Notifications' (bell), and 'Settings' (gear).

**Figure 20 “New Issue” tab: Type of issue selection**

Step 3 – Add Description: The user adds a title and a description that consists of text or voice message, and it is necessary to add at least one photo. It is possible to add more than one photo or a video. He/she then submits the issue.

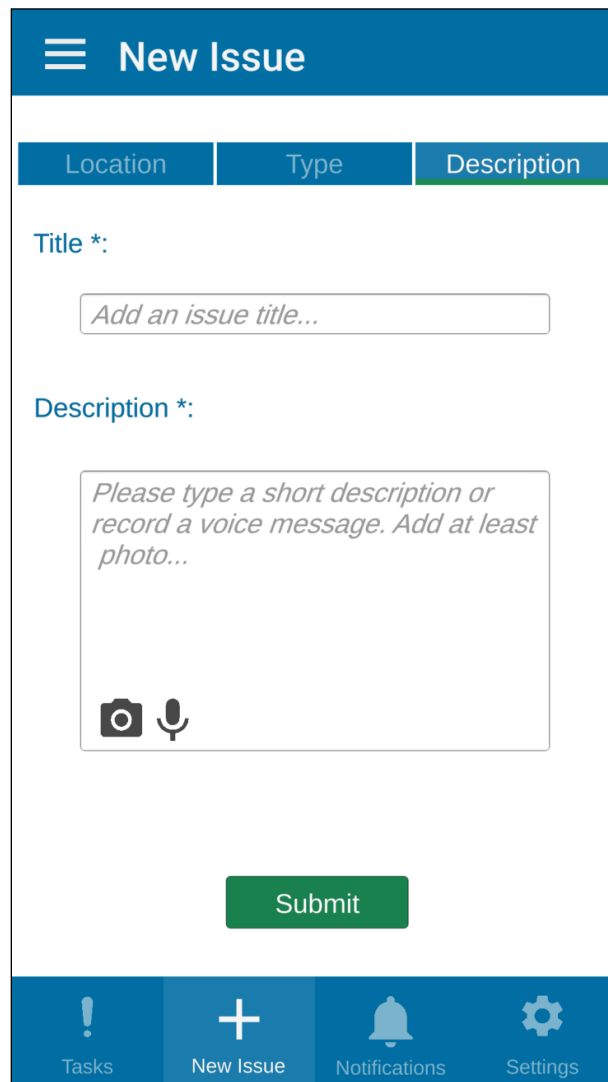
The renovation manager receives the issue and reviews it. He/she replies with acknowledgement or rejection of an issue, and can choose a predefined a text reply or manual text entry.

Possible reply messages, following an indication of acknowledgement or rejection, could be:

- This issue will be considered in the future.
- This issue has been integrated in our schedule under task #000.



- This issue report is incomplete or not clear, please examine the clarity of the content of your form.
- This issue is not acknowledged as an issue.
- Other > “Empty text field”.



**Figure 21 “New Issue” tab: Add Description**

The user will be able to receive notifications on work scheduling that may affect him/her and regarding health and safety (H&S) concerns and reply to them, by completing a form with photographs, videos and text or voice description. These are forwarded to the foreman/project manager etc., to be taken into account for tasks management and scheduling. There are 3 notification types: Scheduling Notifications

(new task), Reminders of Scheduling Notifications (task starts soon) and Task Changes (time range, effects, status etc.)

The user will be able to manage his/her notifications and turn them on/off. Scheduled task notifications:

- Receive Scheduling Notifications prior to a task's scheduling hours including schedule of the upcoming task and Health and Safety information and warnings. – on/off.
- Receive Reminders of Scheduling Notifications on task's scheduling hours including schedule of the task started and Health and Safety information and warnings. – on/off.

Real-time task notifications:

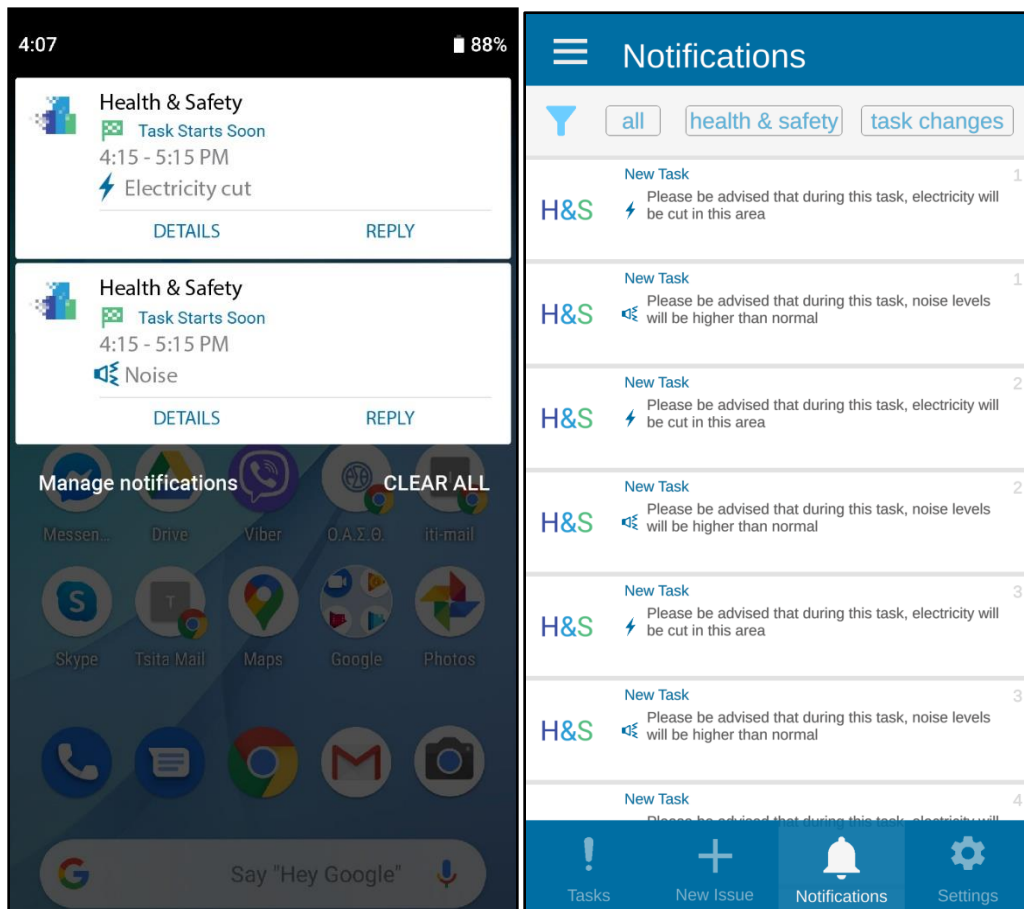
- Receive notifications on every tasks' status and details change – on/off.
  - o Receive notifications only when a task starts and when is finished - on/off.

If it is turned off then the status of tasks info is visible in every task's details, at any time.

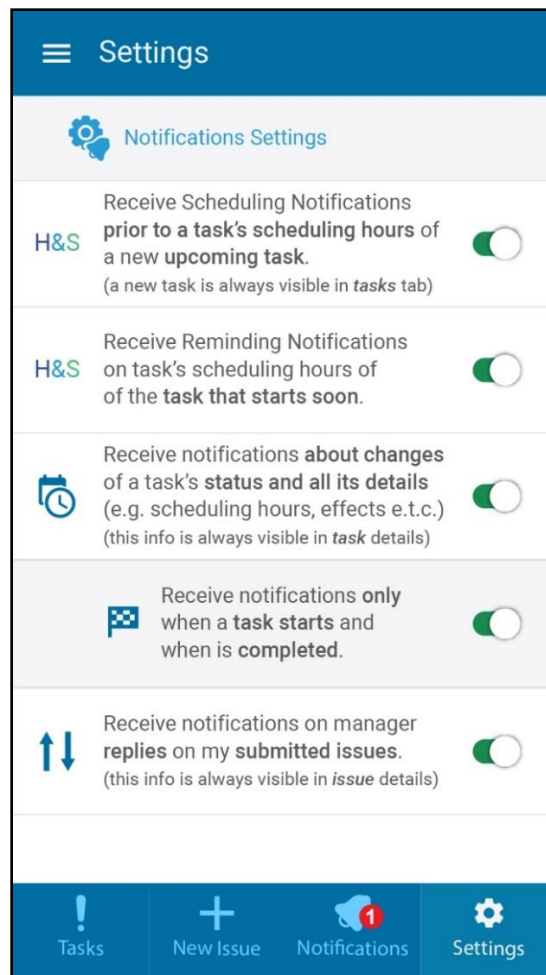
Submitted Issues notifications:

- Receive notifications for manager replies on my submitted issues – on/off

If it is turned off then the status of an issue and info are visible in every issue's details, and will continue to be visible in case the issue becomes a scheduled task. The figure below presents the mock- up of the android notifications and not a real screenshot because this function has not been implemented yet.

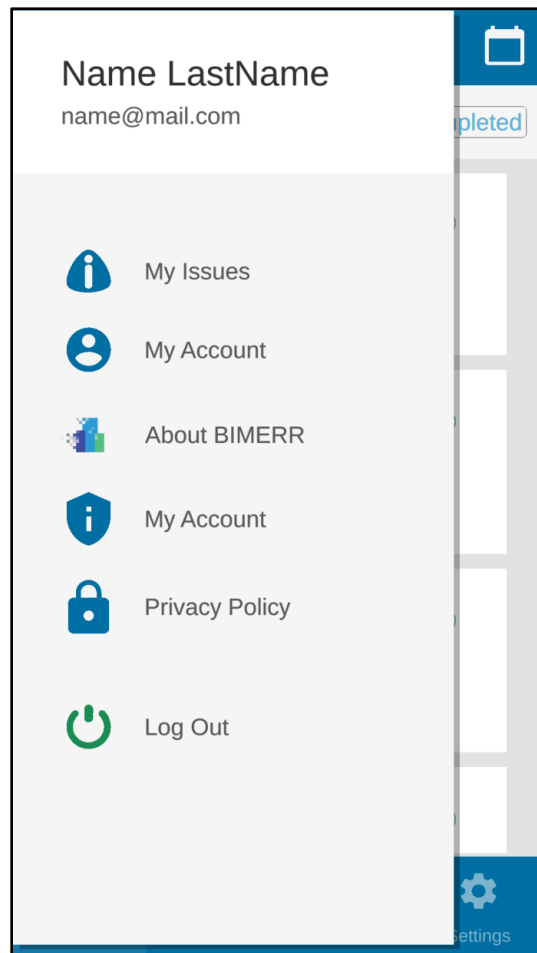


**Figure 22 Android Notifications (Left), In App Notifications (Right)**



**Figure 23 Settings - Notification Settings**

Menu is always directly reachable from a burger button on the top left corner into the first navigation level. The user can enter his/her submitted issues from “My Issues”. The user can enter his/her account info (name, building, mail) and possibly change some of them (“My account”). Enter “About BIMERR” project info, be advised through “FAQs & Tips” on general safety and privacy issues, enter “Privacy Policy” page, or “Log Out”.



**Figure 24 Menu (Navigation Drawer)**

### **5.3 APPLICATION WORKFLOW**

In this subchapter the workflow of the App for residents application is analyzed.

When the user opens the app for the first time, he/she has to register, by filling the input fields with his data and then tapping the register button. The system stores the user's information into a JSON file and sends them for confirmation from the building manager. The user upon his/her registration is prompted to check his/her email for confirmation, which will include a unique code. The user will complete the enrollment by entering the code he/she received in the pop up window which will appear.

After successful authentication, the list of tasks assigned to the logged in user is displayed. The system sends the credentials to the authentication server for confirmation. The back - end process of this procedure will be implemented in the second version because at this time the exact procedure is not determined. When the server approves the user's credential, he will be navigated to the task menu. The tasks menu provides the whole list of tasks that take place only in the building where the user lives, ordered by the expected date and time of execution. From the top right corner the user can switch view mode between "List View" and "Calendar View". The default view is "List View". For displaying the list in this mode, the system retrieves text data in a JSON format from an HTTPS Server (PWMA REST API) and stores them. Then the JSON file is serialized into objects by using the JsonUtility. JsonUtility is utility functions offered by unity for working with JSON data [17]. These objects can be used anywhere during the program execution.

**Figure 25 Retrieved JSON file from PWMA REST API**

```

[Serializable]
20 references
public class Rootobject
{
    public Workorder workorder;
}
[Serializable]
1 reference
public class Workorder
{
    public Header header;
    public Workflow workflow;
    public List<Task> tasks;
}

```

Figure 26 JSON serialization

```

public static Rootobject LoadData(string filename)
{
    string _path = Path.Combine(Application.persistentDataPath, filename);
    string _data = File.ReadAllText(_path);
    return JsonUtility.FromJson<Rootobject>(_data);
}

```

Figure 27 Parsing through JSON Utility

The JSON file includes the tasks of the whole project that the contractor has planned. So there is a strong possibility that tasks are referring to many buildings and not only to the one where the user lives. For this reason, the system checks and displays only the tasks specific to the user's building.

The user can see the progress of each task in the progress bars. The data which exist in the JSON file for these bars, are the expected/actual date and time of execution. The system takes this data and through normalization presents them as progress bars. There are three possible ways to display bars in each task with four flags:

- Two flags for planned start/end - date
- Two flags for real start/end - date

The blue bar is the bar which shows the planned duration, the green one shows the real duration and the orange bar shows how long the task was delayed. These bars have the same fixed size and the normalization depends on the maximum duration between planned end and the real end of the task. The max value of the bar is the max duration and all elements (planned/real start, planned real end of task) are normalized based on this value. The user can also see the types of health and safety issues related to the task. Furthermore, the user can see specific tasks that can have one of the following states according to what view filter he/she has chosen:

- Pending
- Ongoing
- Completed

User can select one task to enter and view its details into a second navigation level in “Task Details”. In this window, he/she can see more details about the task. The system takes this information from the JSON file too. There is an option to also send an annotation about a specific task. This function will be implemented in the second version of the application.

User navigates to the “Calendar View” mode by tapping the calendar icon. In this view, the system is also displaying progress bars of each task but this time with a different approach. In “List View” mode the bars merge in one. In this mode, the bar related to the planned date of execution is separated from the rest. Now the bars show the progress during the month that the user selects. It is worth mentioning that some tasks may be not have been started yet. So the only bar that the user sees is the blue one. Furthermore, “Calendar View” provides the user with the ability to view both the tasks and part of its details on the same screen by clicking on each individual task.

From the tabs menu at the bottom of the screen, the user is able to navigate to “New issue”, “Notifications” and “Settings” screens.

On the New Issue screen, the user is able to report (submit) issues to the AEC project manager or the AEC building maintenance manager, regardless of a scheduled task. He/she is also able to submit potential issues he encounters in the building and make suggestions by completing a form with photographs, videos and text or voice description, at any time. Initially the user sees the Location window with two input fields, one for the apartment and the other for the building which he/she lives. The apartment and building the user is referring to, is already known from his/her account. System takes these data from a JSON file and autofills the input fields. Next step is to choose the type of the issue from a check menu. After the user chooses the type of issue, system takes his choice and stores it into a JSON file too. In the final step he/she describes the issue and also has the option of taking pictures and/or adding video and audio recordings.



The functionality of taking pictures and recordings is not implemented in this iteration as well as the submission of all New Issue's data to manager.

On the Notification screen the user will be able to receive notifications on work scheduling that may affect him/her and regarding health and safety (H&S) concerns and reply to them, by completing a form with photographs, videos and text or voice description. These are forwarded to the foreman/project manager etc., to be taken into account for tasks management and scheduling. There are 3 notification types: Scheduling Notifications (new task), Reminders of Scheduling Notifications (task starts soon) and Task Changes (time range, effects, status etc.) In this iteration the only type of the three that was implemented was the Scheduling notification. System takes the notifications from a JSON file and displays them into the user's Notification screens. System filters the notifications from the JSON file in order to show only the ones that are related to the user. The reply function will also be implemented in the second iteration.

The user will be able to manage his/her notifications and turn them on/off from the Settings screen. In this iteration, only the UI of this Screen was implemented, the back – end functionality will be added in the second iteration.

Menu is always directly reachable from a burger button on the top left corner of the first navigation level. The user can view his/her info by tapping on menu items, such as my issues, my info (name, building, mail), project info, be advised through “FAQs & Tips” on general safety and privacy issues, and finally check the “Privacy Policy” page or “Log Out”. The content of the menu item will be implemented in the second iteration.

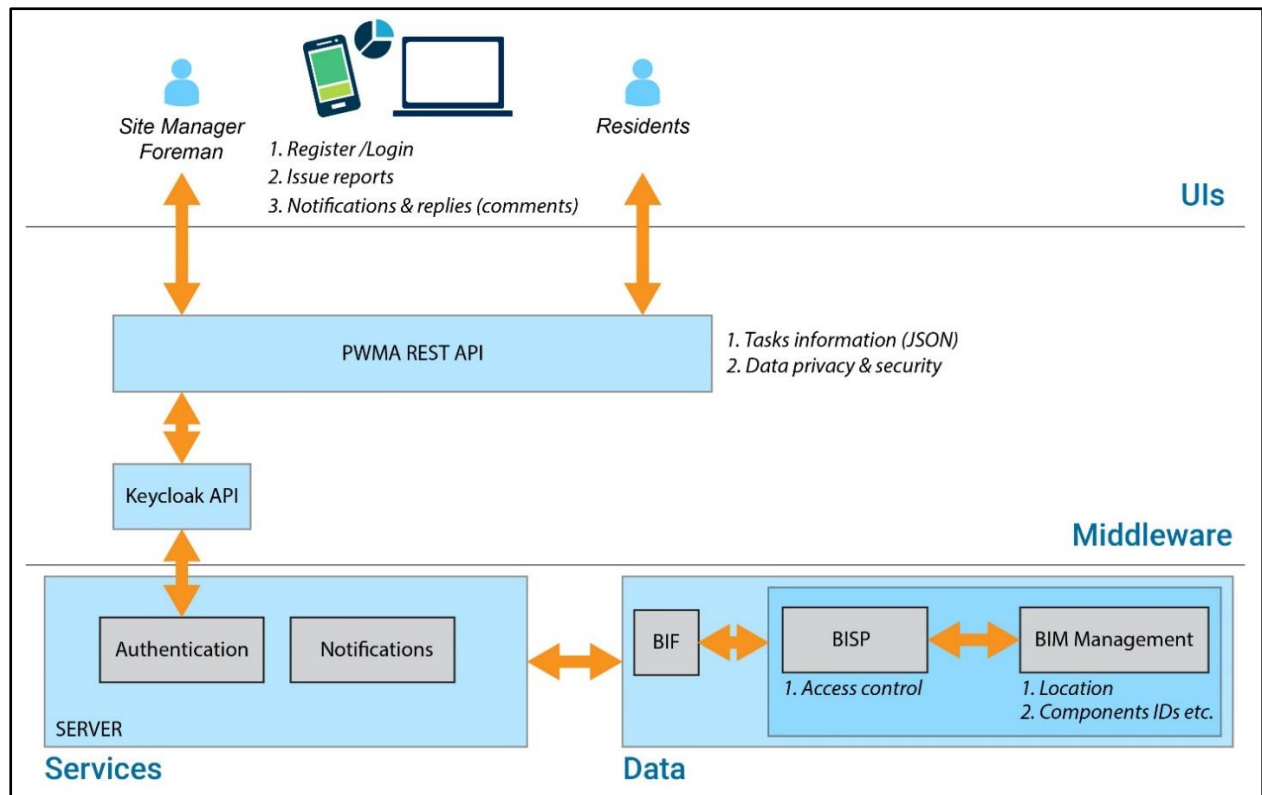


Figure 28 Application workflow

## 4 NEXT STEPS

Next steps include further progress in application design and development as well as in the integration with the rest BIMERR components.

Regarding visualization and content of the GUI design, questions and insights that arose are the following:

- Task details, comments, new issue reports and notification aspects need to be revised in a more detailed level based on data available, others' components processes, back-end definitions, and the pilots' stakeholders' feedback (e.g. definition of location/building/premises in realistic scales convenient to be used and understandable by residents, realistic time ranges and repetitions of task effects, manager replies on new issue reports etc.)
- The use cases (section 3.3), and subsequently information architecture, and navigation need to be revised and validated based on discussion with end-users and in conjunction with pilot specific aspects.
- A question emerged on whether or not it would be safe to be used only by people above a certain age to guarantee the safety of children and youth and the quality of the issue reporting as well. Upon registration, it seems reasonable to recommend the use of the application only by adults or with adult supervision, as the use from children or young adolescents poses safety issues. For example, the functionality of taking photos and videos to include in a report, is considered highly important for the issue assessment by the project manager, but might put the user in danger if close enough to the subject in order to take an accurate photo. This recommendation could be a reminder for all users. What is more is that use only from adults would subsequently be useful to guarantee up to a level the quality of the continues reporting that supports the dialog between residents and AEC project managers. How it will be implemented is still under discussion.
- Settings related to GUI configuration should be defined.

In the next developed version of the application, the following matters need to be tackled:

- One of the next steps that must be done is the communication of the app with the authentication server. In the first iteration, the user can fill the input fields on the Log in and Register screens and system stores the data but doesn't check them against an authentication server. The data stay in the user's device. In addition, in the second iteration the storing of actual voice and video recording as well as camera shots function in the New Issue screen will be added. After this functionality is implemented, consideration will be given about where the data should be sent. At

this moment only the text data are stored in the user's device. One more step that must be done is the push notification.

- The first iteration gives an overview of the main functionalities and the expected interactions of the end-users and poses questions towards exploring limits and possibilities both on front-end and back-end. It is essential that towards the second iteration analytical feedback will be taken and discussions will be held among the partners and with the end-users. Specific pilot approaches will be taken into account as well, for next stages of development and design.

## **5 CONCLUSION AND OUTLOOK**

This deliverable introduces the first version of the application for on-site support of the residents and corresponds to the approach that is described in Deliverable 3.1. The application focuses on creating a direct communication channel between building managers and residents, which can result in enhanced quality of renovation services and the implementation of formative evaluation as input in planning renovation processes, in AEC projects. The provided application is part of a standalone toolset based on the PWMA platform adjusted for the needs of BIMERR.

This document describes the implementation of the first iteration of PWMA App for residents. It analyzes how the design process was laid out. Furthermore the use cases and the user's requirements which are based on deliverable 3.1, are analyzed. Subsequently, it describes the methodology which was followed and the decision of the software support for the implementation of the application. Furthermore, it analyzes the implementation and the features of application as well as the workflow of it. Finally, it defines the next steps which must be done in the second iteration in order for the application to fulfill the specific requirements.

## 6 REFERENCES

- [1] <https://material.io/>, <https://material.io/design/guidelines-overview>
- [2] Chen, J. W., & Zhang, J. (2007). "Comparing Text-based and Graphic User Interfaces for novice and expert users." *AMIA ... Annual Symposium proceedings. AMIA Symposium, 2007*, 125–129.
- [3] Chadwick-Dias, A., McNulty, M. and Tullis, T. (2002). "Web usability and age: how design changes can improve performance". In *Proceedings of the 2003 conference on Universal usability (CUU '03)*. Association for Computing Machinery, New York, NY, USA, 30–37.
- [4] Kascak, L. R., Rébola, C. B., and Sanford, J. A. (2014). "Integrating Universal Design (UD) Principles and Mobile Design Guidelines to Improve Design of Mobile Health Applications for Older Adults," *2014 IEEE International Conference on Healthcare Informatics, Verona, 2014*, 343-348.
- [5] <https://www.w3.org/TR/WCAG21/>
- [6] <https://ionicframework.com/>
- [7] Nayan B. Ruparelia. (2010) "Software development lifecycle models". *SIGSOFT Softw. Eng. Notes* 35, 3 (May 2010), 8–13. DOI:<https://doi.org/10.1145/1764810.1764814>
- [8] Wynn, D.C., Clarkson, P.J. (2018). "Process models in design and development". *Res Eng Design* 29, 161–202. <https://doi.org/10.1007/s00163-017-0262-7>
- [9] Larman, C. and Basili, V. R. (2003). "Iterative and Incremental Development: A Brief History". *Computer* 36, 6 (June 2003), 47–56. DOI: <https://doi.org/10.1109/MC.2003.1204375>
- [10] Wodtke, C. (2002). "Information Architecture: Blueprints for the Web"
- [11] Pruitt, J. and Adlin, T. (2010). "The persona lifecycle: keeping people in mind throughout product design", Morgan Kaufmann
- [12] Beck, K., Grenning, J. J., Martin, R. C., Beedle, M., Highsmith, J., Mellor, S., Bennekum, A., Hunt, A., Schwaber, K., Cockburn, A., Jeffries, R., Sutherland, J., Cunningham, W., Kern, J., Thomas, D., Fowler, M., Marick, B. (2001). "Principles behind the Agile Manifesto". Agile Alliance. Archived from the original on 14 June 2010. Retrieved 6 June 2010.
- [13] <https://unity.com/>
- [14] <https://www.microsoft.com/en-us/hololens>
- [15] <https://docs.microsoft.com/en-us/dotnet/csharp/>
- [16] <https://visualstudio.microsoft.com/>
- [17] <https://docs.unity3d.com/2018.4/Documentation/Manual/index.html>

[18] <https://www.json.org/json-en.html>

[19] <https://uxplanet.org/how-to-create-personas-step-by-step-guide-303d7b0d81b4>