



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 D6.11

Renovation progress monitoring & alerting application for residents 2

Deliverable Status: **Final**
 File Name: **BIMERR_D6.11-v1.00.docx**
 Due Date: **09/06/2021 (M30)**
 Submission Date: **30/06/2021 (M30)**
 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

The BIMERR 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

Disclaimer

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	Poveda-Villalón	María	UPM	mpoveda@fi.upm.es
2	Lucerski	Maciej	BX	maciej.lucerski@budimex.pl

TABLE OF CONTENTS

<i>List Of Tables</i>	7
<i>List Of Figures</i>	8
<i>Acronyms</i>	12
<i>Executive Summary</i>	14
1 Introduction	15
1.1 Objectives of the Deliverable	15
1.2 Changes to the First Version of the PWMA for Residents Application	15
1.3 Design, Methodology and Implementation	16
1.3.1 Design Process	16
1.3.2 User Requirements and Usage Scenarios	16
1.3.3 Development plan and Technical Requirements	17
1.3.4 Implementation, Features and Application Workflow	17
1.3.5 API Documentation	18
1.3.6 User Manual.....	18
2 Design Process	19
2.1 PWMA Application for Residents' main goal and functionalities	19
2.2 GUI Design Approach.....	24
2.3 GUI Design Methodology.....	26

3	<i>User Requirements and Usage Scenarios.....</i>	29
3.1	Occupants’ requirements	29
3.1.1	Requirements	29
3.1.2	Personas and use case stories	31
3.1.3	Information Architecture & Navigation	35
3.1.4	Design Choices – Layout	40
3.2	Usage Scenarios	44
3.2.1	US – 01: Notifications and residents’ responses related to tasks	44
3.2.2	US – 02: Residents’ responses related to a specific task	45
3.2.3	US – 03: Residents’ new issue reports	45
4	<i>Development plan and Technical Requirements</i>	47
4.1	Development Plan.....	47
4.2	Development Toolset	48
4.2.1	Unity	48
4.2.2	C#	49
4.2.3	Visual Studio	49
5	<i>Implementation, Features and Application Workflow.....</i>	51
5.1	Implementation	51
5.2	Features.....	53
5.3	Application Workflow	67

6	<i>Integration With The Other BIMERR components</i>	74
6.1	Task Data & Notifications	74
6.2	Push Notifications & Registration Verification	75
6.3	Issue Reporting & Tasks Comments	77
7	<i>Conclusion And Outlook</i>	80
8	<i>Bibliography</i>	81
	<i>Annex I – User Manual</i>	83
	Description on How To Use/Operate	83
	Troubleshooting	107

LIST OF TABLES

Table 1 BIMERR Use Case 2	20
Table 2 BIMERR Use Case 3	20
Table 3 BIMERR Use Case 10	21
Table 4 BIMERR Use Case 11	21
Table 5 BIMERR Use Case 12	22
Table 6 BIMERR Use Case 13	22
Table 7 BIMERR Use Case 15	23
Table 8 Main Principles of Universal Design	25

LIST OF FIGURES

Figure 1 PWMA Application for residents General Architecture.....	17
Figure 2 Iterative Design & Development Methodology	27
Figure 3 Iteration circle of Design & Development Methodology	28
Figure 4 Home page “Tasks”- List view (left), Calendar view (center), and Menu (Navigation Drawer) (right).....	36
Figure 5 Information Architecture.....	38
Figure 6 Navigation Diagram – Mock-up screens	39
Figure 7 Mock-ups, Colors and Indicators Home page “Tasks” - List view (left), Calendar view (right)	41
Figure 8 Mock-ups, Android Notifications (left), “Notifications” page (center), “Settings”- Notifications Settings (right)	42
Figure 9 Mock-Ups, Selected task – Details of tasks (second navigation level)	43
Figure 10 Mock-Ups, “New Issue” page: Location confirmation (Left), Type of issue selection (Center), Add Description (Right).....	44
Figure 11 Unity environment.....	49
Figure 12 Example of triggering the Main Menu	52
Figure 13 Example of the GUI which remains relative in every aspect ratio	53
Figure 14 Login screen	54
Figure 15 Register screen (left), pop up window for finalizing registration (right).....	55

Figure 16 Task tab – list view	56
Figure 17 Selected task – Task Details, into the second navigation level.....	57
Figure 18 Tasks Tab – Calendar View	58
Figure 19 “New Issue” tab: Location confirmation.....	59
Figure 20 “New Issue” tab: Type of issue selection	60
Figure 21 “New Issue” tab: Add Description	61
Figure 22 Android Notifications (Left), In App Notifications (Right).....	63
Figure 23 Settings - Notification Settings.....	64
Figure 24 Menu (Navigation Drawer)	65
Figure 25 My Issues	66
Figure 26 About BIMERR	67
Figure 27 Retrieved JSON file from BIF REST API.....	69
Figure 28 JSON serialization.....	69
Figure 29 Parsing through JSON Utility	70
Figure 30 Application workflow	73
Figure 31 BIF upload to API endpoint	74
Figure 32 UnityWebRequest Example Code.....	75
Figure 33 Android Notifications Example	76

Figure 34 Annotation Data collection setup 1	77
Figure 35 Annotation Data collection 2	77
Figure 36 Annotation Data collection setup 3	78
Figure 37 Annotation Data collection setup 4	78
Figure 38 Annotation data model example	79
Figure 39 Installation Procedure 1	84
Figure 40 Installation Procedure 2	85
Figure 41 Installation Procedure 3	86
Figure 42 Register Screen	87
Figure 43 Registration verification	88
Figure 44 Login Screen	89
Figure 45 Tasks List Screen	90
Figure 46 Tasks List Screen (2)	91
Figure 47 Task Details Screen	92
Figure 48 Camera Capture/Confirmation	93
Figure 49 Video Capture/Confirmation	94
Figure 50 Pick from Gallery	95
Figure 51 Calendar View Screen	96

Figure 52 Calendar View Screen 2.....	97
Figure 53 New Issue Screen 1	98
Figure 54 New Issue Screen 2.....	99
Figure 55 New Issue Screen 3.....	100
Figure 56 Android Notifications (Left), In App Notifications (Right).....	101
Figure 57 Comment a Notification	102
Figure 58 Notifications Settings Screen	103
Figure 59 Menu Screen	104
Figure 60 My Issues Screen.....	105
Figure 61 About BIMERR	106
Figure 62 Log Out	107

ACRONYMS

Acronym	Meaning
AEC	Architecture Engineering Construction
API	Application Programming Interface
AR	Augmented Reality
VR	Virtual Reality
ARIBFA	Augmented Reality enabled In-situ Building Feature Annotation
BICA	Building Information Collection Application
BIF	BIMERR Interoperability Framework
BIM	Building Information Model
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
SDK	Software Development Kit
UC	Use Case
UUID	Universally Unique Identifier
WCAG	Web Content Accessibility Guidelines

EXECUTIVE SUMMARY

This document analyses the second version of the mobile application for on-site support of the residents and corresponds to the approach that is described in Deliverable 3.1 “Stakeholder requirements for the BIMERR system” [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.

While the first version focused on covering the user requirements, the GUI design and the implementation of the basic UI functionalities, the second version focuses on covering the improvements and the finalization of all the previous steps and the deeper integration with other components of the BIMERR ecosystem to exchange valuable data. This document provides details about the finalized versions of the user requirements and use cases, the Graphical User Interface (GUI) design, the development plan and the technical requirements. Moreover it provides a detailed description about the implementation and the application workflow, the Application Programming Interfaces (APIs) that are needed for the integrations with the other BIMERR components and a complete User Manual.

1 INTRODUCTION

1.1 OBJECTIVES OF THE DELIVERABLE

This deliverable introduces the second version of the mobile application for on-site support of the residents. 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. 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 second version of the application aims at integrating 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 and the BIMERR Interoperability Framework.

1.2 CHANGES TO THE FIRST VERSION OF THE PWMA FOR RESIDENTS APPLICATION

In the Deliverable 6.10, submitted in M20, a number of functionalities and improvements were remarked to be included in the second version of the PWMA for residents application. In this section, the functionalities added (or modifications made) to the initial version of the tool are summarised:

- Integration with Keycloak: The user can now log in after Keycloak sends back to him/her the access token.
- Registration verification. The user can now register after receiving a verification code.
- Integration with BIF:
 - Sending annotations with actual voice and video recordings as well as photographs using “New Issue” or “Task details” Screen.
 - Receiving actual work order data from BIF
- Receive push notifications from the PWMA notification server
- Implementation of the rest screens and functionalities such as “My Issues” screen.
- GUI finalization: Aspect ratio relativity, proper fonts, colours and icons.

1.3 DESIGN, METHODOLOGY AND IMPLEMENTATION

1.3.1 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 (interviews, questionnaires, brainstorming etc.) and technical specifications. 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 fed the development of the first version of the application (Chapter 2). Next iterations incorporated that feedback to expand the application development both in detail and content, but also to revise previously made design choices.

1.3.2 User Requirements and Usage Scenarios

The user requirements specify what users expect from the system in terms of its functionalities. User Requirements were written before the system was created. The results from the questionnaires analysis were mapped onto BIMERR user requirements. The prioritization of end users requirements was also produced, setting the initial guidelines for the design and development of the PWMA for residents D3.1 Section 6 [1].

A usage scenario is a list of actions describing the steps, events, and/or actions which occur during the interaction. Based on the proposed Business Scenarios [1], usage scenarios were identified. The aim was to describe a list of system usage scenarios that highlight the main principles of the PWMA for residents and hence, an appropriate subset can be tested during the pilot trials. The usage scenarios (section 3.3) were defined after consultation both with the technical and the pilot partners of the project.

1.3.3 Development plan and Technical Requirements

After the design process was finished and the resulting mock ups were at hand, the next phase was to plan the implementation process. This process was 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 (Chapter 4). The architecture covers the general functionalities of the application as well as the "ecosystem" of external sources with which information is exchanged during the project. The general structure of the application's architecture is presented in figure 1.

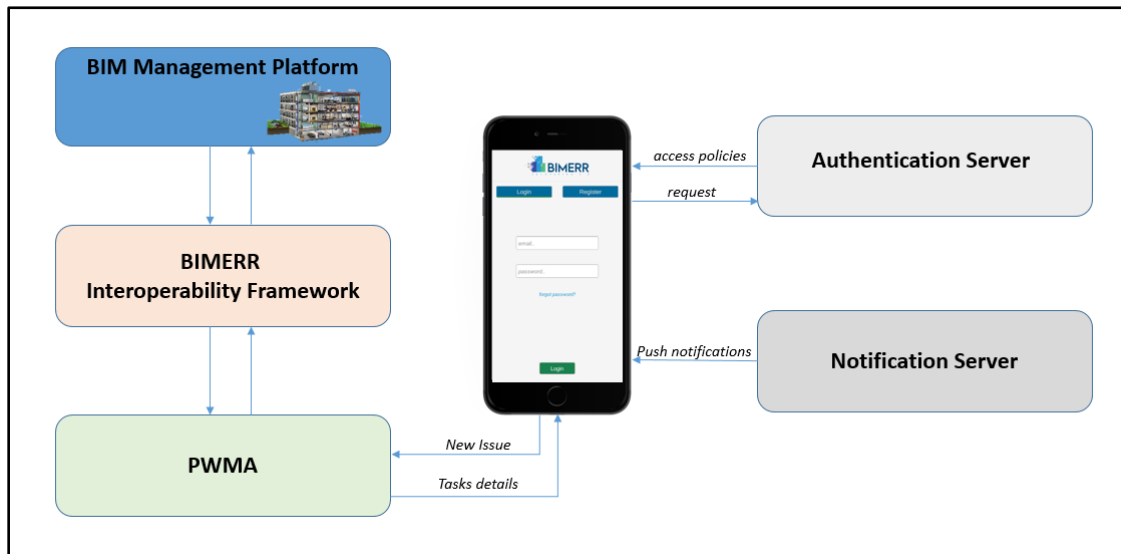


Figure 1 PWMA Application for residents General Architecture

1.3.4 Implementation, Features and Application Workflow

The implementation of the application begun 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 PWMA for residents will be described, as well as the workflow process which is a series of sequential tasks that were 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. Implementation, Features and Application Workflow.

1.3.5 API Documentation

For the integration with other BIMERR's components, the application needed to be able to send and receive data such as the tasks of the workorders, notifications, issue reporting etc. For these actions PWMA for residents had to use APIs for communication with BIF and PWMA back-end. In Chapter 6 the integration with these BIMERR components will be described.

1.3.6 User Manual

For the use of the application guidelines and instructions are essential. Following the final version implementation of the toolset, the users have to be able to be informed about the way that the app works and the features which offers. In chapter 7 description details on how to use the application and details of the User Interface are provided as well as a section regarding how to troubleshoot any accessibility problem.

2 DESIGN PROCESS

The Graphical User Interface (GUI) design process started 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 followed 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 followed an iterative and incremental design process. Feedback on a design circle highlights gaps and inconsistencies that fed a new circle, which started 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 aspired to solve. In this way, the flow of actions and the components chosen for the visualization of the main screens of the application was defined, 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.

2.1 PWMA APPLICATION FOR RESIDENTS' MAIN GOAL AND FUNCTIONALITIES

In order to understand the services that 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 involved the user group Building Occupants.

The following tables include those use cases that involve the Building Occupants user group D3.1 Section4. 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:

Deliverable D6.11 ■ 06/2021 ■ CERTH

Page 19 of 108

BIMERR project ■ GA #820621

Table 1 BIMERR Use Case 2

BIMERR Use Case 2	
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"> • The residents input is important, as at times they have irreplaceable knowledge. • 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. • Input directly on BIM would be a more professional tool, and thus not easy or meaningful to be used by everyone.

Table 2 BIMERR Use Case 3

BIMERR Use Case 3	
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"> • 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.

Table 3 BIMERR Use Case 10

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"> • Maybe not all occupants would be interested to get involved or fully understand the works performed and why. Occupants are not always owners. • 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. • Regarding the feedback and the introduction of slight changes from the occupants should be expressed through a simple way and as they perceive it.

Table 4 BIMERR Use Case 11

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"> • Health and Safety (H&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.
--	--

Table 5 BIMERR Use Case 12

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 related aspects	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 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.
Notes	<ul style="list-style-type: none"> • 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. • They should be able to report any issue and at any time through a simple and easy-to-use interface.

Table 6 BIMERR Use Case 13

BIMERR Use Case 13	
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.
Notes	<ul style="list-style-type: none"> • 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.

Table 7 BIMERR Use Case 15

BIMERR Use Case 15	
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.
Notes	<ul style="list-style-type: none"> 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.

All in all, PWMA for residents 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 renovation 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 for residents application, 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 was 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, seemed 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 for residents application 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 Building Information Collection Application (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 for residents 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 for residents application main Functionalities (by priority order):

- 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.

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.

2.2 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 was to make it as simple as possible. As well it had to be 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 was important to address usability issues. Design guidelines such as Web Content Accessibility Guidelines (WCAG) 2.1¹ are embedded in numerous development frameworks such as Ionic^{2 3} etc.

Although, the GUI design followed material design for all BIMERR mobile applications, it was 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 could be reviewed to avoid risks and inconsistencies.

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”.

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

Table 8 Main Principles of Universal Design

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

¹ <https://www.w3.org/TR/WCAG21/>

² <https://material.io/>, <https://material.io/design/guidelines-overview>

³ <https://ionicframework.com/>

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

2.3 GUI DESIGN METHODOLOGY

There are many design and development process models (e.g. waterfall, V-model etc.) [5], 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 (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 [6].

A generic iterative design process had 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 did 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 have been risky and unnecessary. The second reason was that PWMA for residents application is heavily depended on other modules and applications, ordering for a highly flexible design process that would have allowed constant flow of information and feedback among consortium partners. Nonetheless, not much was changed on the second version regarding the content and layout of the main screens that were designed and developed for the first version [7].

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

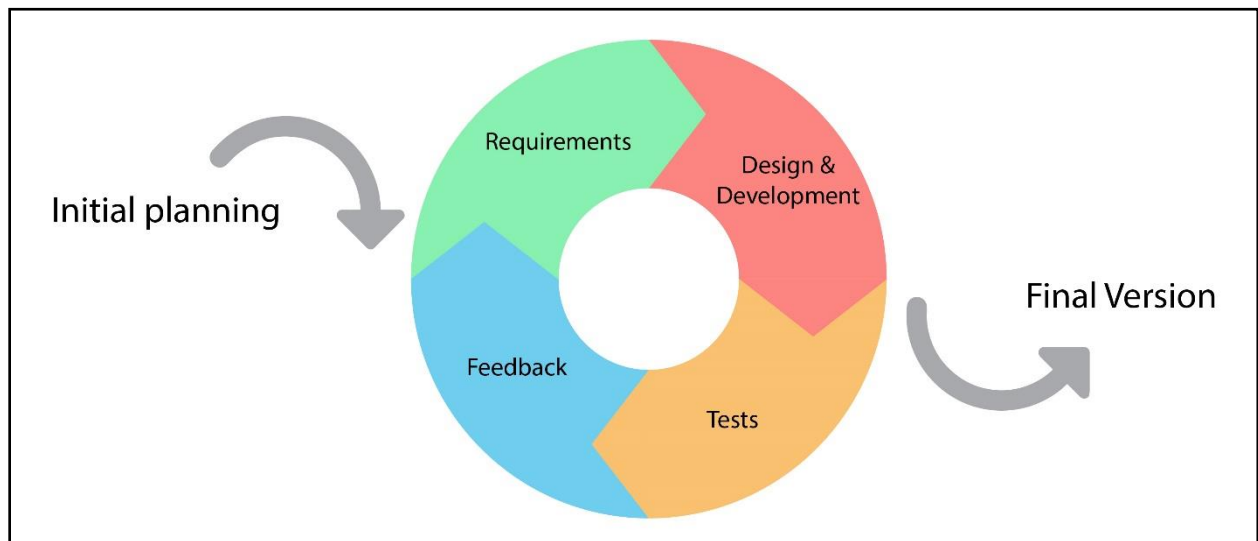


Figure 2 Iterative Design & Development Methodology

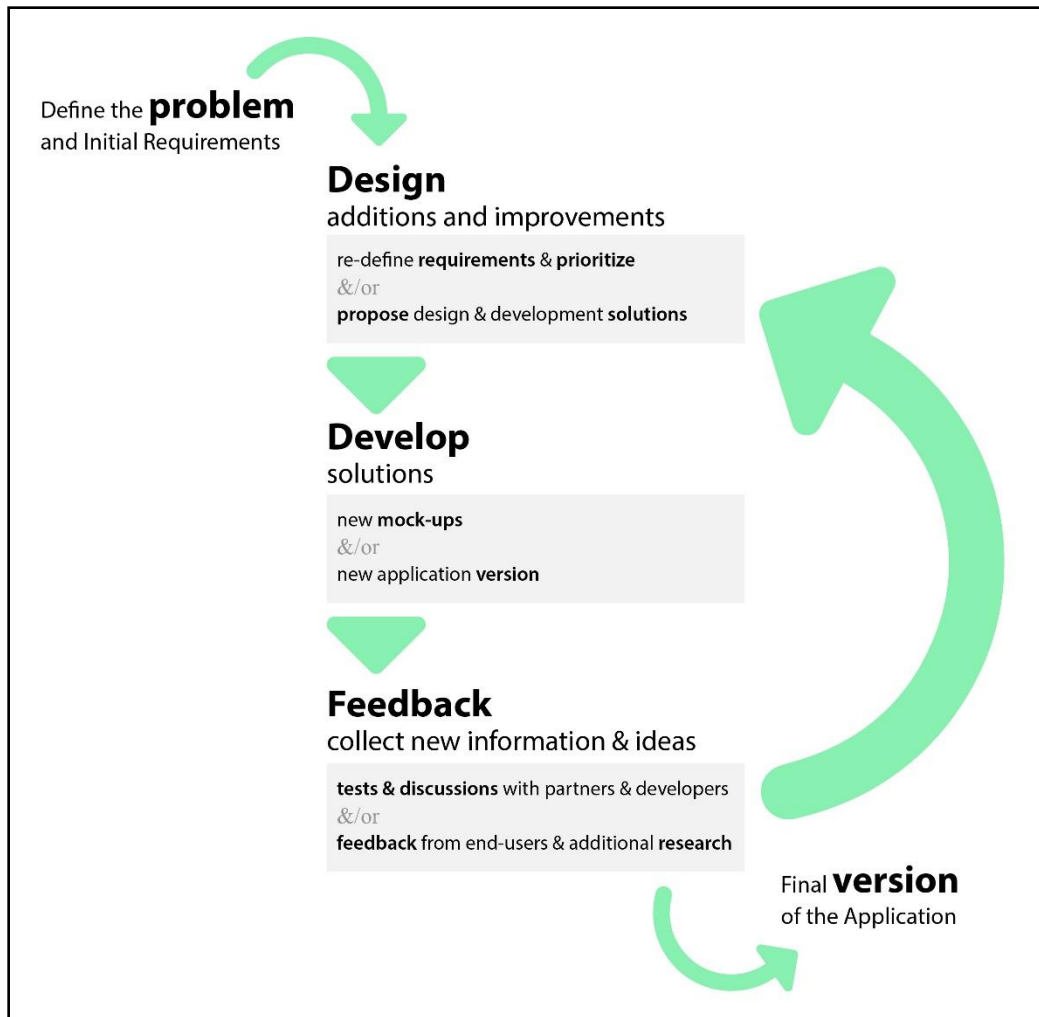


Figure 3 Iteration circle of Design & Development Methodology

3 USER REQUIREMENTS AND USAGE SCENARIOS

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 for residents application (section 3.2).

Based on those, as a first step – design, CERTH designers team used Personas⁴ 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 led 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, were taken into account and were prioritized based on feedback, discussion and further research among the partners and the end-users. That feedback and additional information collection was the last step of the first iteration and the outcome fed the last iterations (Figure 3).

3.1 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

3.1.1 Requirements

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

Users

The application for residents is used by the user group “Building Occupant” and by the “Building Manager”. Secondly, this application could 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

⁴ <https://uxplanet.org/how-to-create-personas-step-by-step-guide-303d7b0d81b4>

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 is used via any android-based tablet/mobile phone.
- The application for residents is 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.

3.1.2 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.

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 2 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 resident 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 were 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 taken 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: Plumbing 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

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.

3.1.3 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, ongoing, 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 (Figure 4).

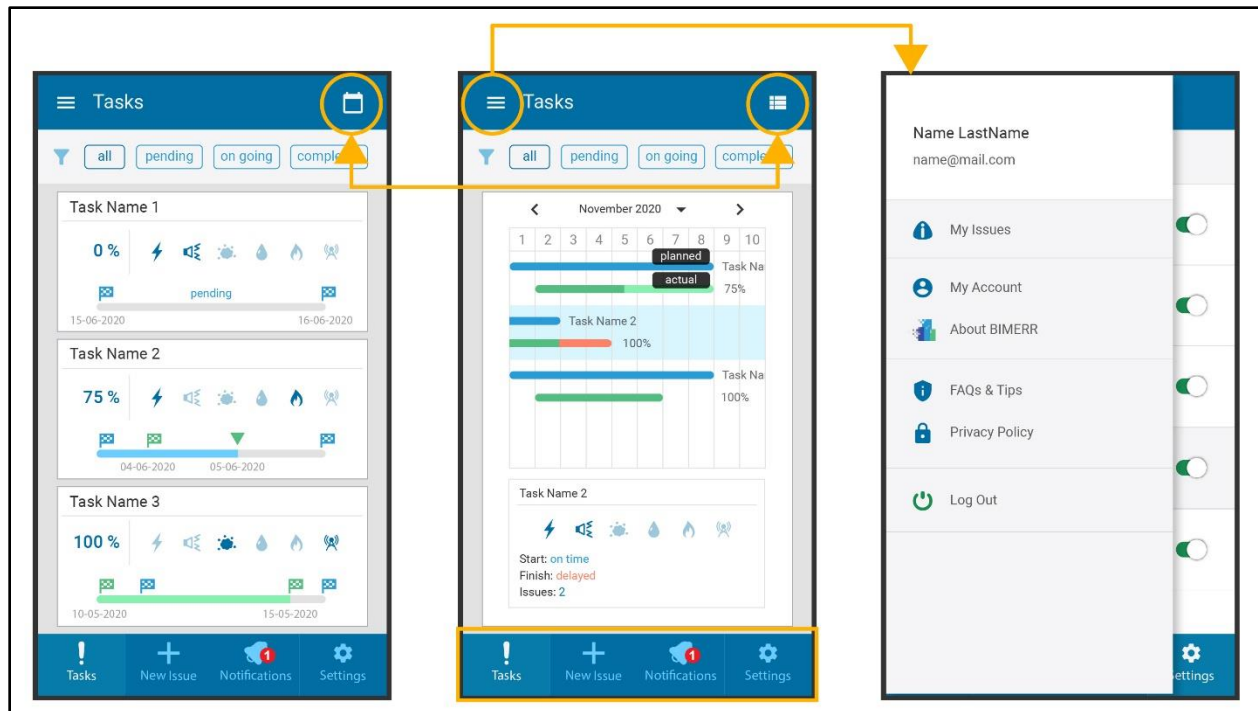


Figure 4 Home page “Tasks”- List view (left), Calendar view (center), and Menu (Navigation Drawer) (right)

As seen in the Information Architecture (Figure 5), there are four main tabs (grey rectangles): “Tasks” – screens indicated with numbers 3 and 4 in the navigation diagram (Figure 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 (Figure 4). In both views, the user can use the filter to view on of the tasks’ categories: all, pending, ongoing, 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 (Figure 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 ((Figure 6, screen no.6), 2 – select type of issue (Figure 6, screen no.7), 3 – adds description including text, video, photos and/or voice message (Figure 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.

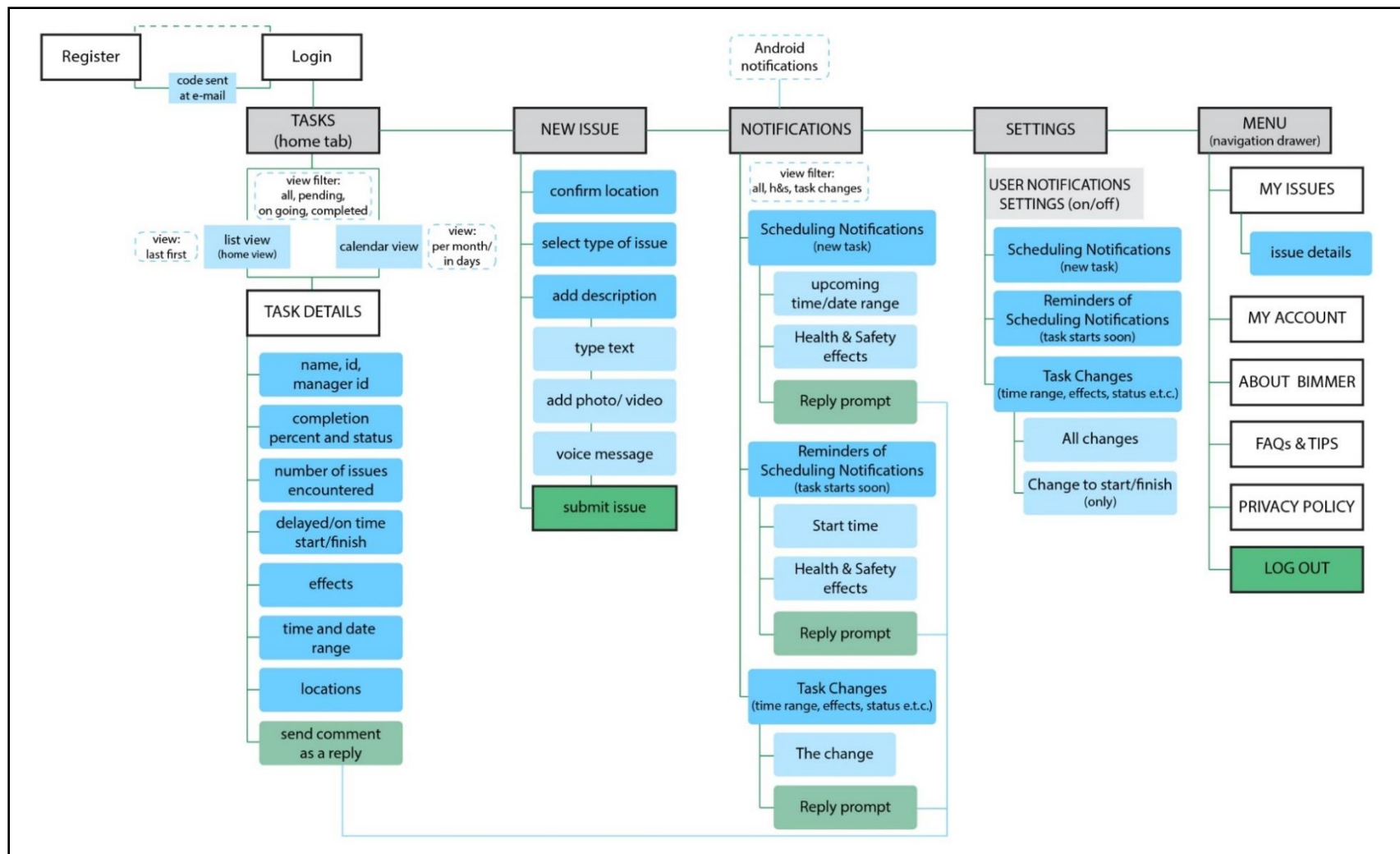


Figure 5 Information Architecture



Figure 6 Navigation Diagram – Mock-up screens

3.1.4 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 guidelines⁵ and the respective criteria. All icons, texts and even layout have been redefined as the content and functionalities continued 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).

⁵ <https://www.w3.org/TR/WCAG21/>

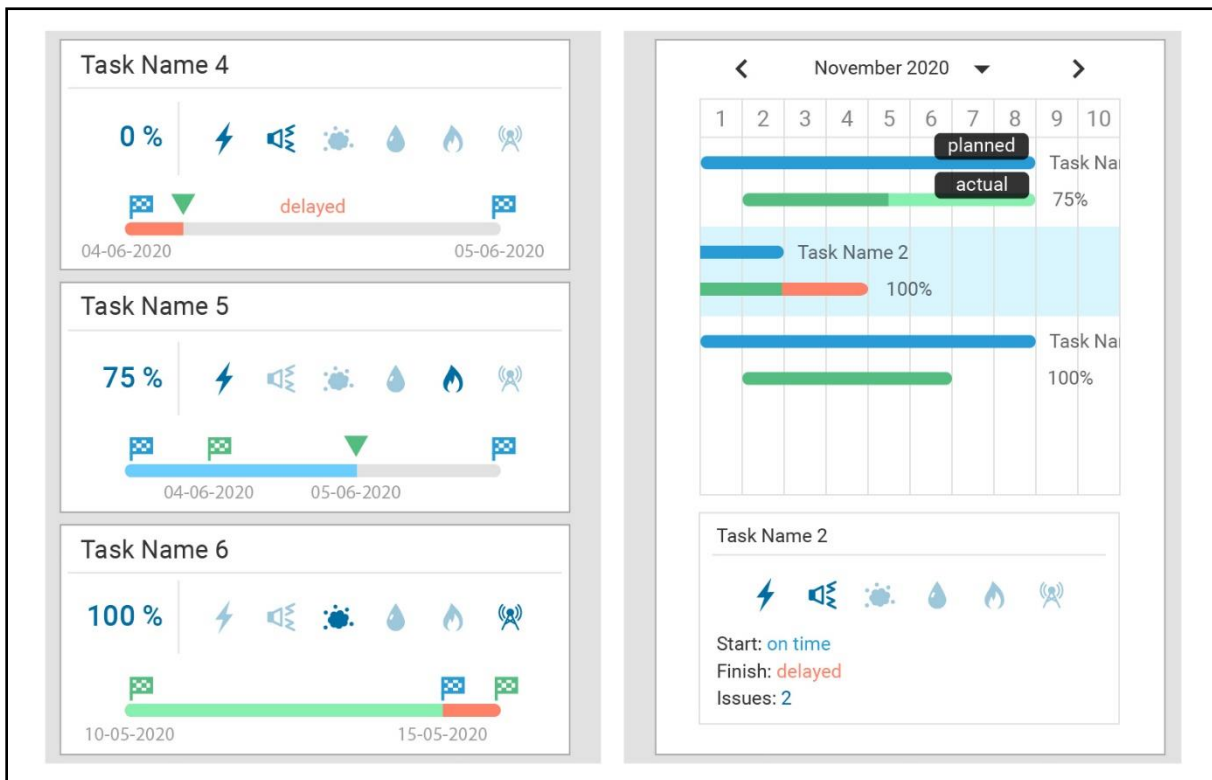


Figure 7 Mock-ups, Colors and Indicators Home page “Tasks” - List view (left), Calendar view (right)

- 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) (Figure 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 taping on a list item of the tasks' list view mode.

- Direct access to any last change is essential for the user and for that reason 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) (Figure 8).

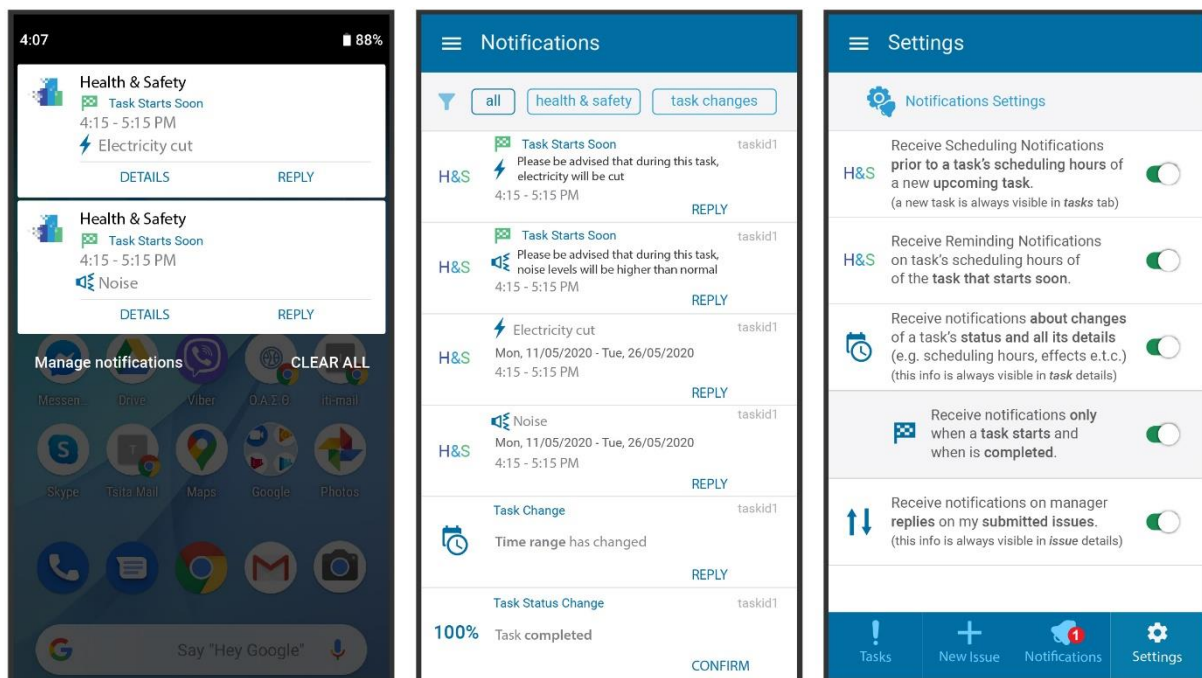


Figure 8 Mock-ups, Android Notifications (left), "Notifications" page (center), "Settings"- Notifications Settings (right)

- 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).

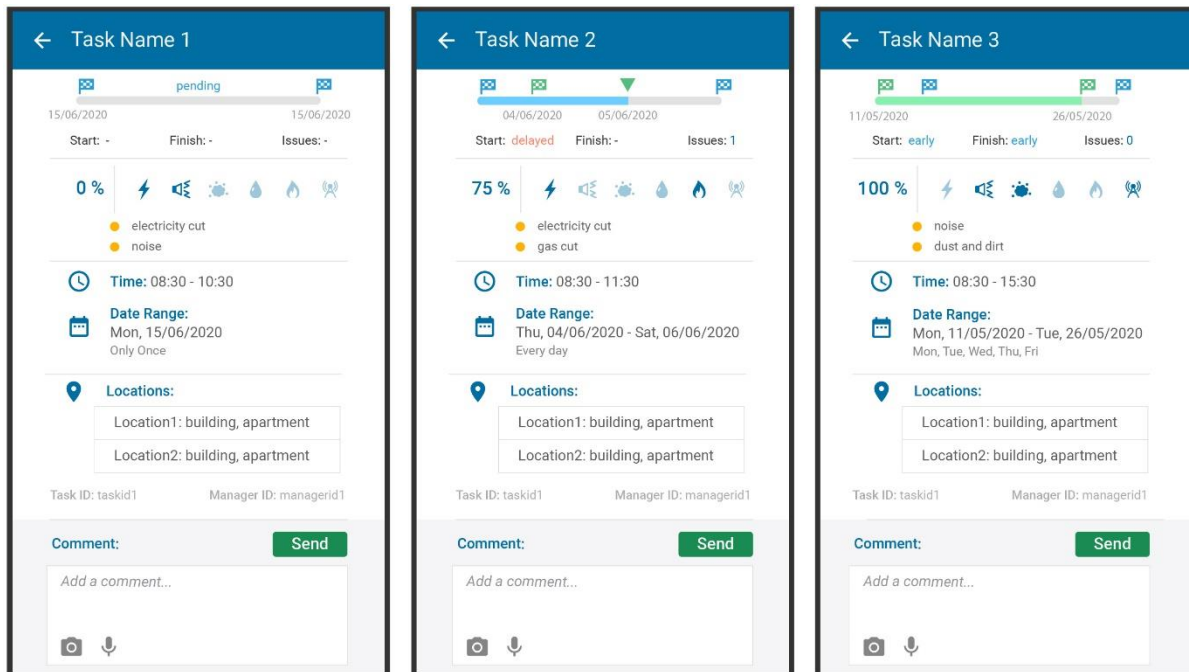


Figure 9 Mock-Ups, Selected task – Details of tasks (second navigation level)

- 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 (Figure 10).

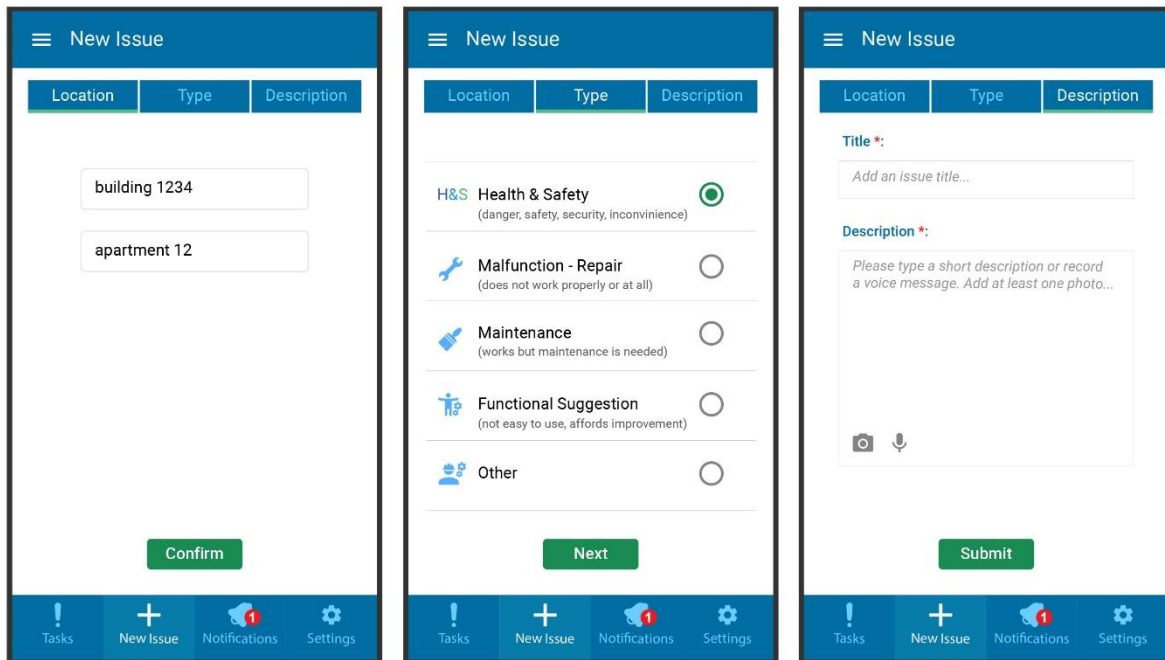


Figure 10 Mock-Ups, “New Issue” page: Location confirmation (Left), Type of issue selection (Center), Add Description (Right)

3.2 USAGE SCENARIOS

Based on the personas scenarios, and the layout that is presented above, the main flow of users interactions with PWMA for residents application led to the usage 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 usage scenarios.

3.2.1 US – 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.

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.

3.2.2 US – 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.

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.

3.2.3 US – 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.

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 took 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 [8].

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. Unity 2018.4.22 was the version that was chosen for App for residents

Unity⁶ 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 (Figure 11). 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. Since 2018 Unity had been used to create 60 percent of augmented reality and virtual reality content, including the emerging augmented reality platforms, such as Microsoft HoloLens⁷.

⁶ <https://unity.com/>

⁷ <https://www.microsoft.com/en-us/hololens>

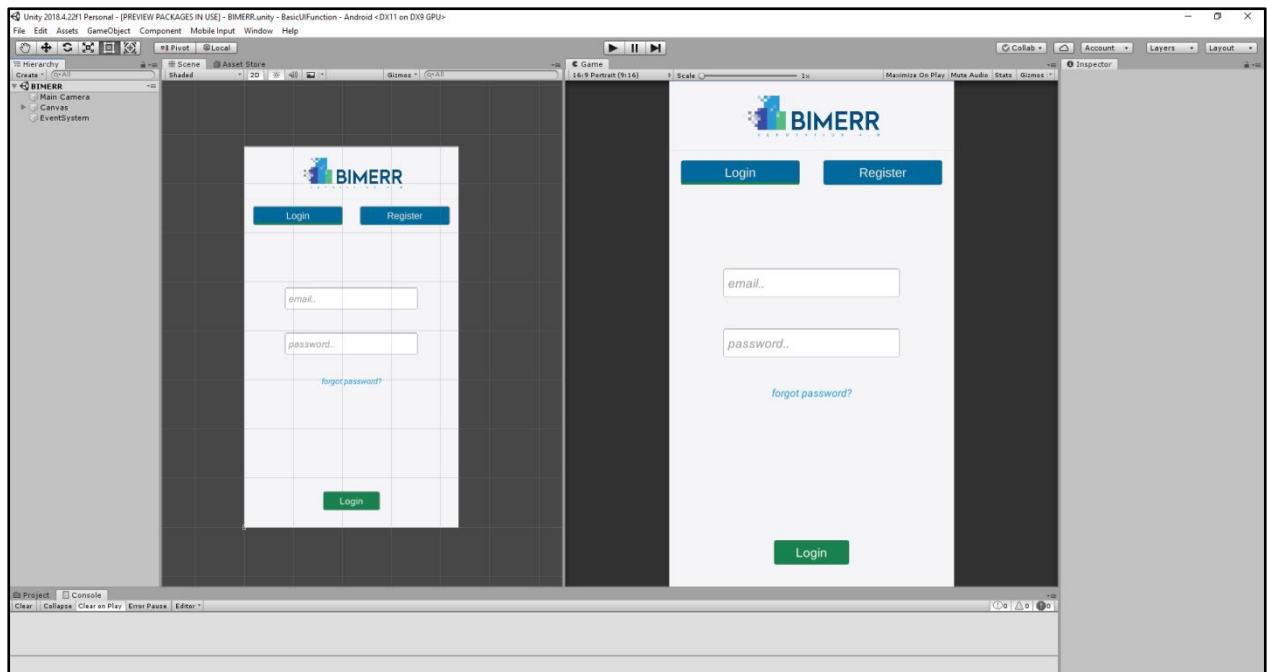


Figure 11 Unity environment

4.2.2 C#

As mentioned above Unity offers the object-oriented programming language C#⁸. 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).

4.2.3 Visual Studio

Visual Studio⁹ 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

⁸ <https://docs.microsoft.com/en-us/dotnet/csharp/>

⁹ <https://visualstudio.microsoft.com/>

supports smart code completion techniques and code refactoring. It supports almost all programming languages, including C #, which is used to create scripts in Unity.

5 IMPLEMENTATION, FEATURES AND APPLICATION WORKFLOW

In this chapter the implementation process, which came 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 UI¹⁰ 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 as 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 12). 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 was 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 13).

¹⁰ <https://docs.unity3d.com/2018.4/Documentation/Manual/index.html>

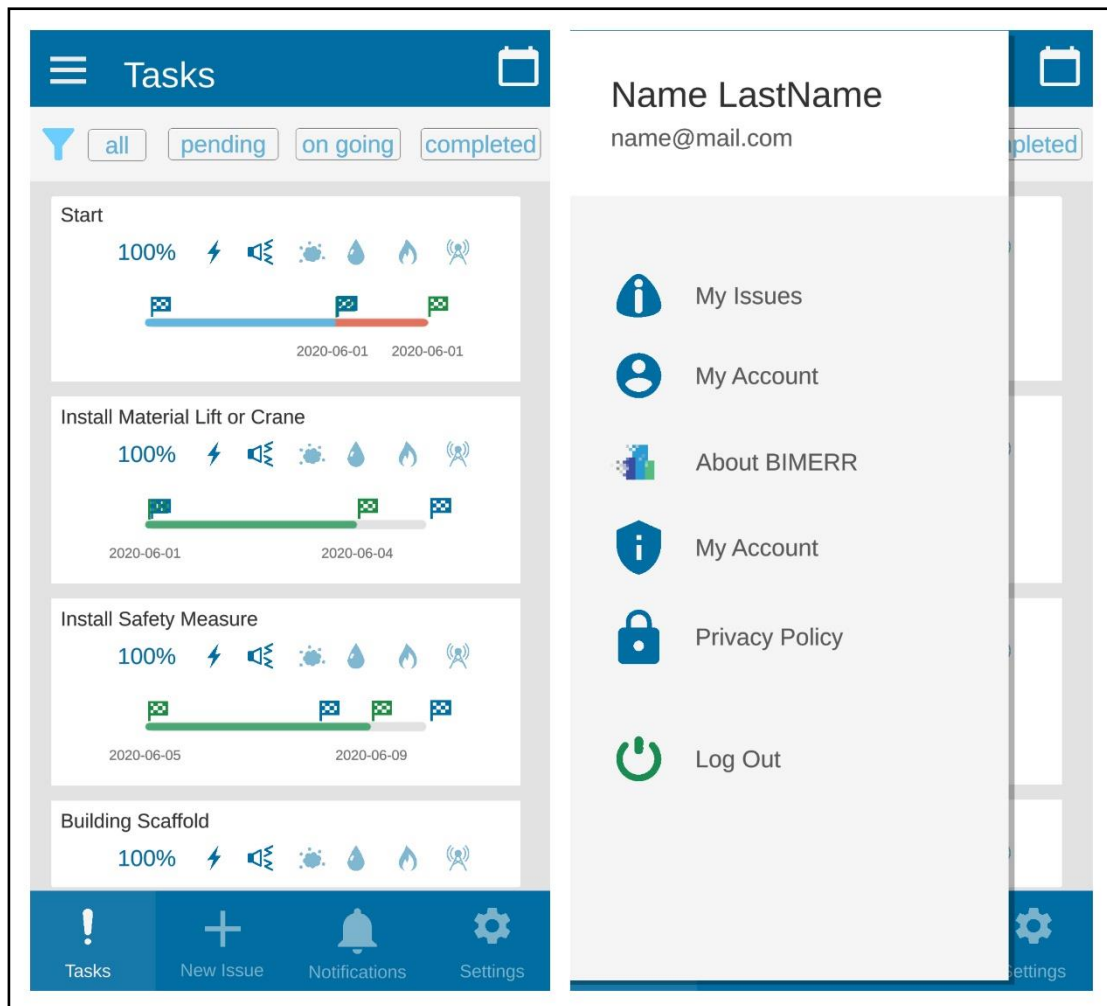


Figure 12 Example of triggering the Main Menu

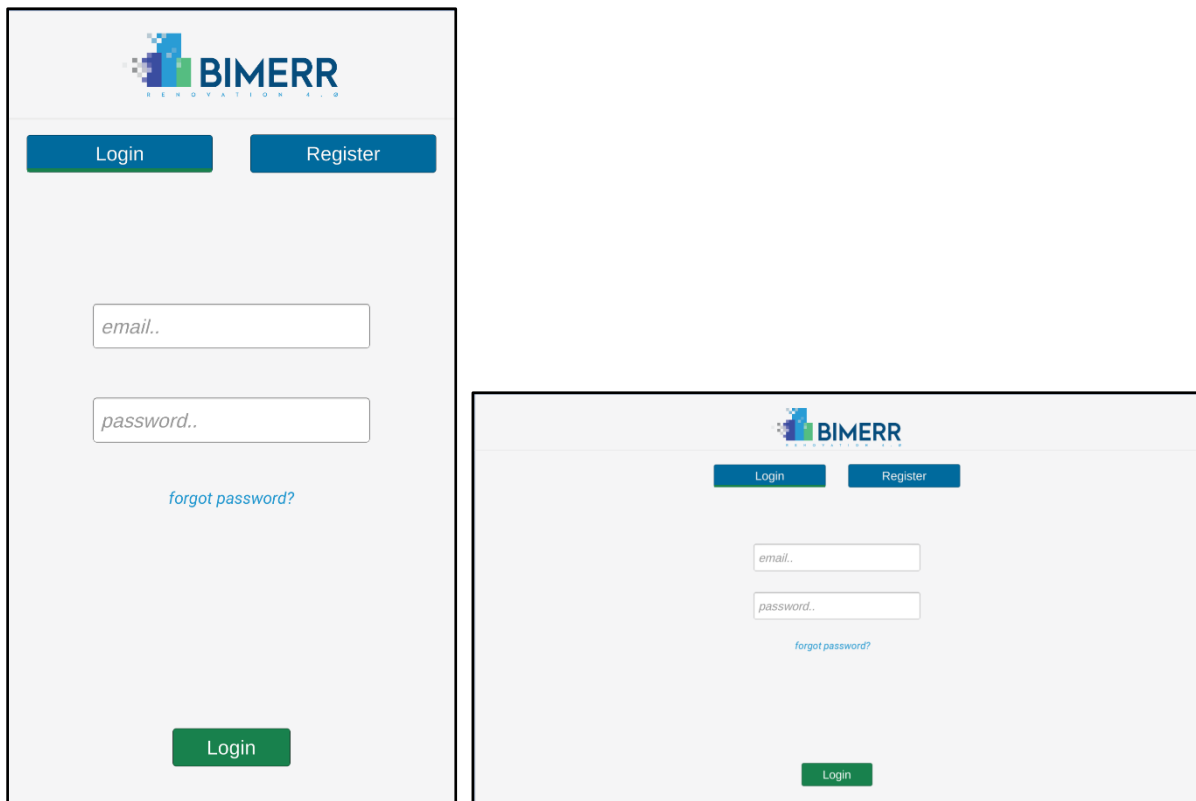


Figure 13 Example of the GUI which remains relative in every aspect ratio

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)¹¹ file format which is a readable format for structuring data. Correspondingly, the data related to user, such as task details are received in JSON file format and parsed. The parsing and the generating of them is easy to be executed by the machines.

5.2 FEATURES

All features have already been implemented in the second version, and they are fully functional after the integration with all the other modules of BIMMER.

¹¹ <https://www.json.org/json-en.html>

The first screen is the login window, which asks the user to log in with his/her assigned username and password (Figure 14). 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.

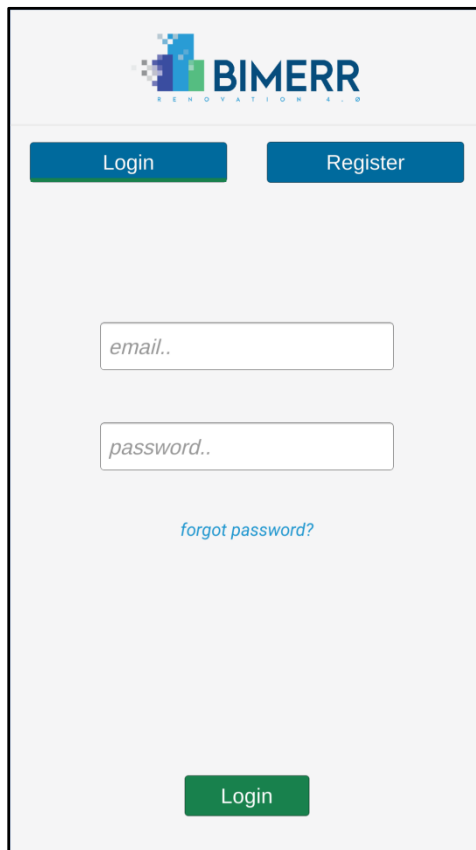
The image shows a mobile application login screen for BIMERR. At the top, there is a logo with a colorful square icon and the text 'BIMERR' in blue, with 'RENOVATION' in smaller text below it. Below the logo are two blue buttons: 'Login' and 'Register'. In the center, there are two white input fields with placeholder text 'email..' and 'password..'. Below these fields is a blue link that says 'forgot password?'. At the bottom, there is a green button labeled 'Login'.

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 (Figure 15). The user will complete the enrollment by entering the code he/she received in the pop up window which will be prompted.

The figure displays two screenshots of the BIMERR application interface. The left screenshot shows the main registration form, which includes the BIMERR logo at the top, followed by 'Login' and 'Register' buttons. Below these are input fields for 'first name', 'last name', 'phone number (optional)', 'email..', 'password', 'building', and 'apartment'. A green 'Register' button is positioned at the bottom of the form. The right screenshot shows a 'Complete Registration' pop-up window. It features the BIMERR logo, 'Login' and 'Register' buttons, and a message: 'Please check your email for the unique code.' Below this is an input field labeled 'enter code..', a green 'Register' button, and a blue link that reads '[I have not received an e-mail.](#)'. At the bottom of the pop-up, there is a faded 'apartment' input field and a faded 'Register' button.

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 (Figure 16). Furthermore, user can see specific tasks that are pending, ongoing or completed according to what view filter is chosen.

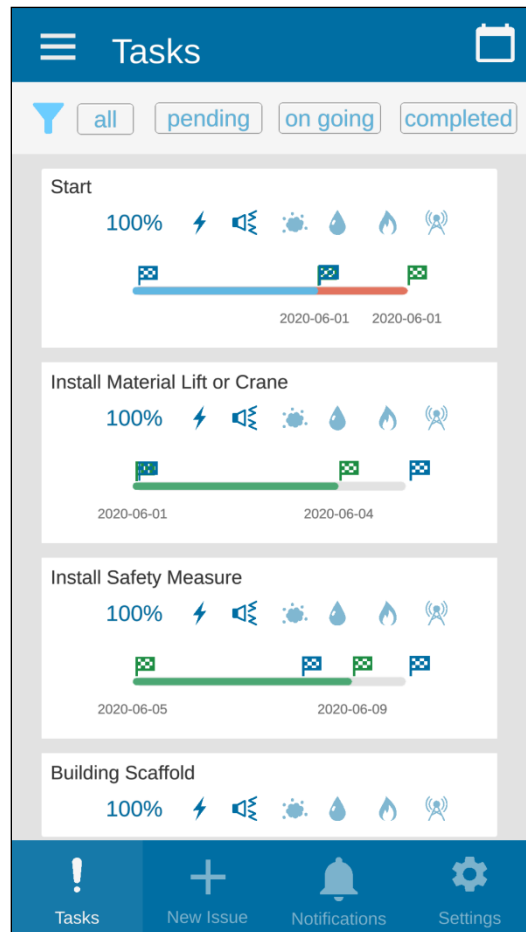


Figure 16 Task tab – list view

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).

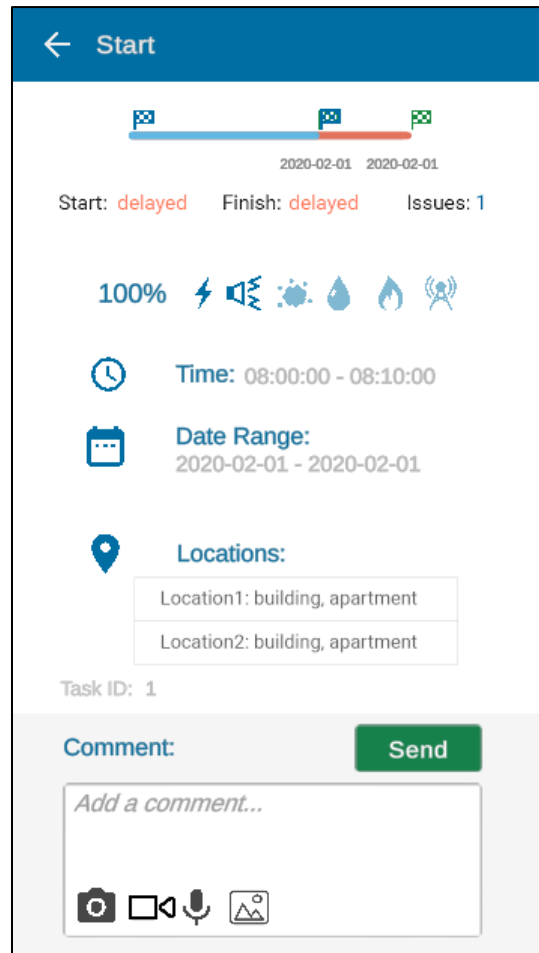


Figure 17 Selected task – Task Details, into the second navigation level

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 18 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.

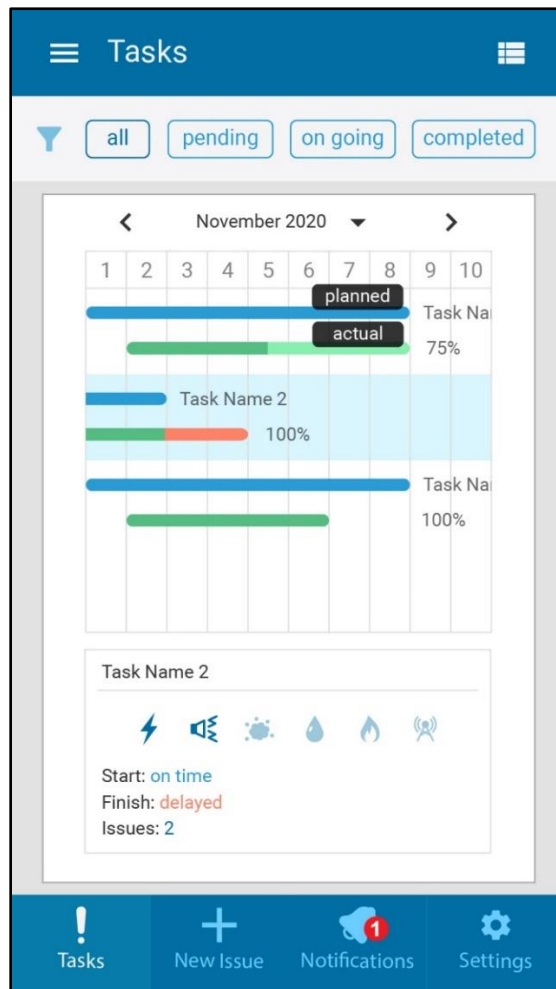


Figure 18 Tasks Tab – Calendar View

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 (Figure 19).

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 (Figure 20).

- 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. The 'Functional Suggestion' row has a green radio button, indicating it is selected. Below the table is a green 'Next' button. At the bottom is a blue navigation bar with four icons and labels: 'Tasks' (exclamation mark), 'New Issue' (plus sign), 'Notifications' (bell), and 'Settings' (gear).

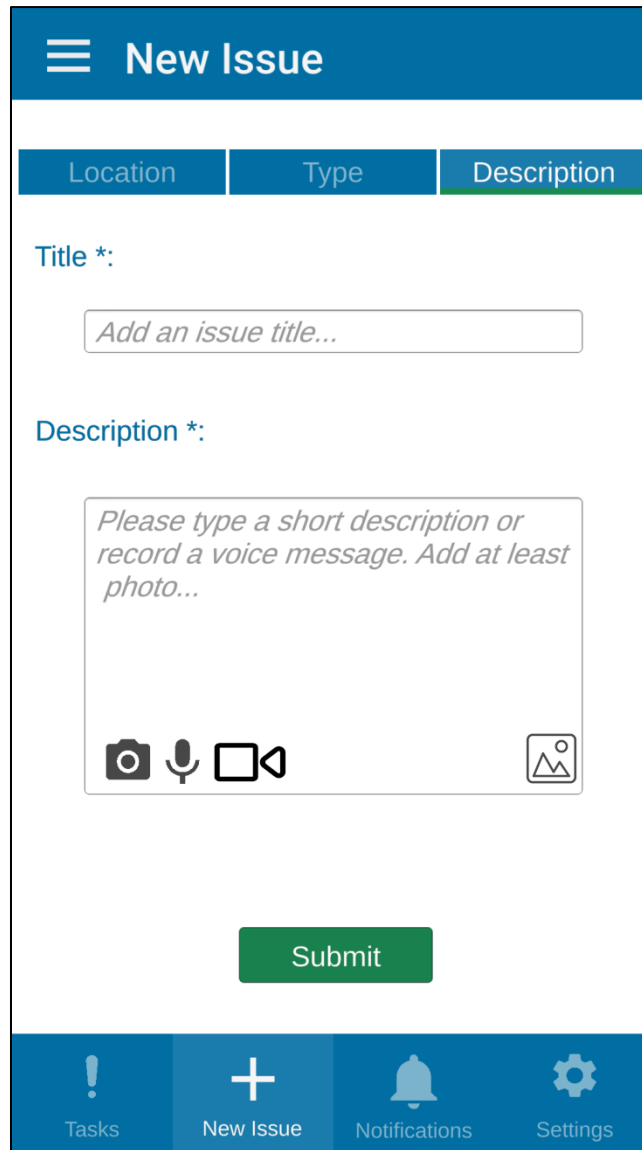
Location	Type	Description
H&S	Health & Safety (danger, safety, security, inconvenience)	<input type="radio"/>
	Malfunction - Repair (does not work properly or at all)	<input type="radio"/>
	Maintenance (works but maintenance is needed)	<input type="radio"/>
	Functional Suggestion (not easy to use, affords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

Next

Tasks New Issue Notifications Settings

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

Step 3 – Add Description: The user adds a title and a description that consists of a text or voice message, and it is necessary to add at least one photo (Figure 21). It is possible to add more than one photo or a video and he/she can view the photographs, videos and audio recordings before their submission. He/she then submits the issue.



New Issue

Location	Type	Description
<p>Title *:</p> <input type="text" value="Add an issue title..."/>		
<p>Description *:</p> <div> <p>Please type a short description or record a voice message. Add at least photo...</p> <div> </div> </div>		
<p>Submit</p>		

Tasks

New Issue

Notifications

Settings

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, send comments to the related tasks by tapping on them and navigating to the “task details” screen of the related task when he/she can send photographs, videos and a 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.) (Figure 22).

The user will be able to manage his/her notifications and turn them on/off (Figure 23). 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.
 - 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.

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.

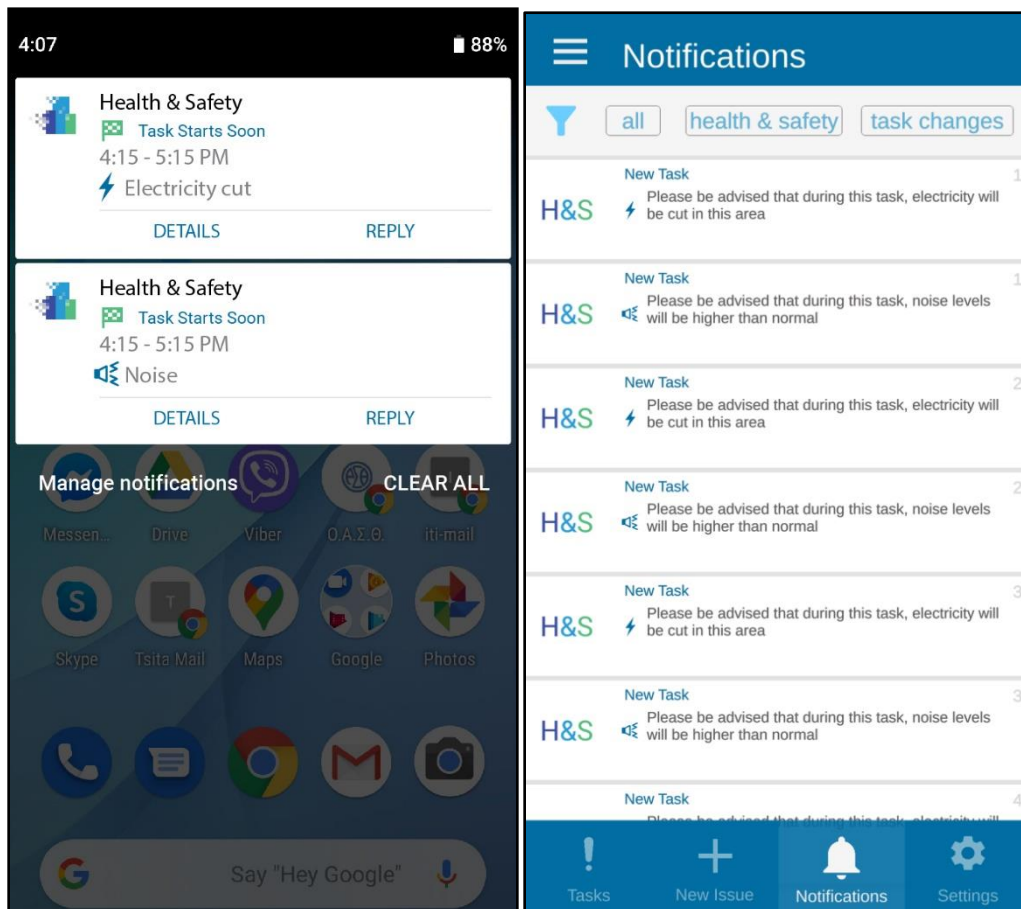


Figure 22 Android Notifications (Left), In App Notifications (Right)

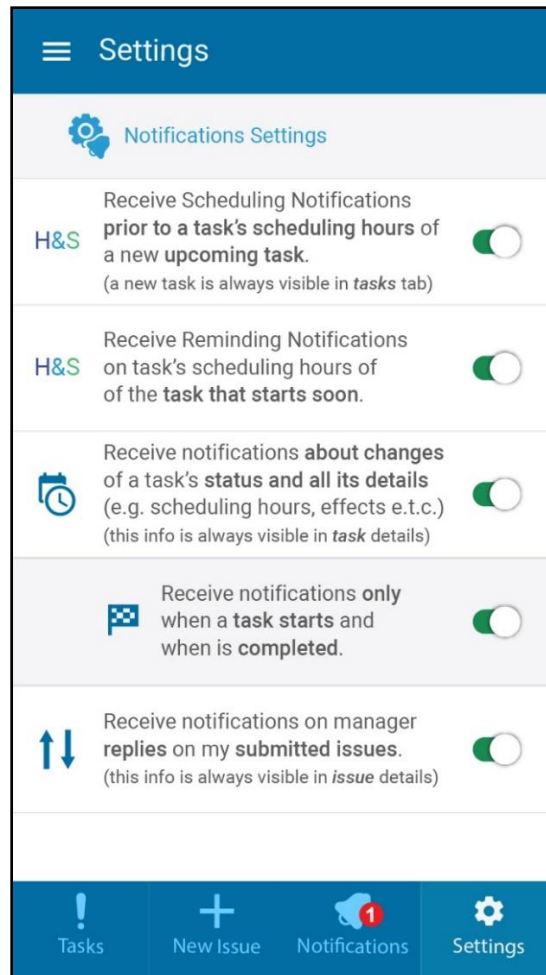


Figure 23 Settings - Notification Settings

Menu is always directly reachable from a burger button on the top left corner into the first navigation level (Figure 23). The user can:

- see his/her submitted issues from “My Issues” (Figure25). On this screen user is able to see all the submitted issues’ details. He/she can see the submitted text and the photos, play the videos, listen to audio recordings that he/she has sent.
- check his/her account info (name, building, mail) and possibly change some of them (“My account”).

- read “About BIMERR” project info. The user can read general information about PWMA for Residents and BIMERR and he/she can tap on a link which navigates him/her to the BIMERR webpage (Figure 26).
- be advised through “FAQs & Tips” on general safety and privacy issues,
- read about the privacy policy on “Privacy Policy” page,
- log out”.

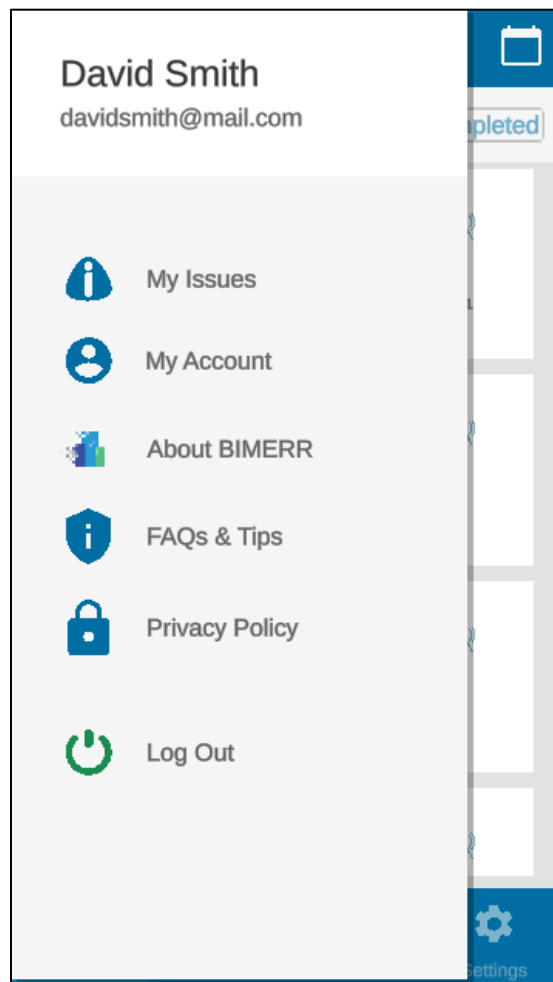


Figure 24 Menu (Navigation Drawer)

←

My Issues

AC ISSUES

Location:



Building: A, Apartment: 12

Type:

Malfunction repair

Description:

The AC has stopped working.

CIRCUIT BREAKER ISSUES

Location:




Building: A, Apartment: 12

Type:

Health & Safety

Description:

The circuit breaker smells like it is burning.

POWER OUTAGE

Location:

Building: A, Apartment: 12

Type:

Health & Safety

Description:

I have a power outage in my apartment.

Figure 25 My Issues



Figure 26 About BIMERR

5.3 APPLICATION WORKFLOW

In this section the workflow of the App for residents application is analysed.

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 to the building manager. To make this possible, the system sends the data

through a channel of the Notification server in which the building manager has subscribed to the channel. The Building manager receives the registration data as push notification in his/her own PWMA for residents account which offers additional features. More specifically the system enables an extra screen to the building manager's account with a list for the apartments of the building. The building manager has to crosscheck the registration data that he/ she will receive through push notification from the notification server with the information that he/she has of each individual resident. If the building manager approves the user's credentials he/ she will send a Universally Unique Identifier (UUID) to the user through an API channel of the notification server. This UUID is a unique code which is derived from a function which generates it based on the zone ID where the user lives. The user upon his/her registration is prompted to check his/her push notifications, which will include this uuid code. The user will complete the enrollment by entering the code that he/she received in the pop up window which will appear. The system has to check if the received UUID is correct by decoding it. After the successful registration (and every time when the user has logged out) he/she has to log in by filling in his/her email and his/her password. The system sends the credentials to the Keycloak authentication server for confirmation and the Keycloak returns an access token to the user. After successful authentication, he will be navigated to the task menu where the list of tasks assigned to the logged in user is displayed.

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 BIMERR Interoperability Framework (BIF) through BIF API and stores them (Figure 27). Then the JSON file is serialized into objects by using the JsonUtility (Figure 28, 29). JsonUtility¹² is utility functions offered by Unity for working with JSON data. These objects can be used anywhere during the program execution.

¹² <https://docs.unity3d.com/ScriptReference/JsonUtility.html>

```
{
  "workorder": {
    "header": {
      "id": 1,
      "name": "Test:BIMERR RECONSTRUCTION 001",
      "description": "Test:Facade Improvement Template",
      "start": {
        "planned_start": "2020-06-18 22:00:00",
        "actual_start": null,
        "finish": {
          "planned_finish": "2020-09-18 22:00:00",
          "actual_finish": null
        }
      },
      "workflow": {
        "header": {
          "id": 1,
          "name": "Facade Improvements - Outside of the Wal",
          "building_id": 379930,
          "project_id": "process_obj.58134"
        },
        "tasks": [
          {
            "id": 1,
            "name": "Start",
            "description": "obj.58004| Start for Start",
            "execution_time": {
              "planned": {
                "id": 51,
                "workorder_result_id": 1,
                "logattributes_id": 1,
                "value": 5,
                "comments": null,
                "start": "2020-06-01 08:00:00",
                "end": "2020-06-01 08:10:00",
                "real": {
                  "id": 52,
                  "workorder_result_id": 1,
                  "logattributes_id": 1,
                  "value": 6,
                  "comments": null,
                  "start": "2020-06-01 08:10:00",
                  "end": "2020-06-01 08:15:00"
                }
              },
              "costs": {
                "planned": {
                  "id": 1003,
                  "workorder_result_id": null,
                  "logattributes_id": 2,
                  "value": 4,
                  "comments": "mockup"
                },
                "real": {
                  "id": 1002,
                  "workorder_result_id": null,
                  "logattributes_id": 2,
                  "value": 3,
                  "comments": "mockup"
                }
              },
              "locations": [
                {
                  "type": "noise",
                  "notes": "Please be advised that during this task, electricity will be cut in this area"
                },
                {
                  "type": "noise",
                  "notes": "Please be advised that during this task, noise levels will be higher than normal"
                }
              ],
              "building_components": [
                {
                  "id": "sdgskgmfsdg545ldfd",
                  "type": "Wall",
                  "name": "Wall13484"
                },
                {
                  "id": "dknfdk49h301525hfjhd",
                  "type": "DoorFrame",
                  "name": "DoorFrame36"
                }
              ],
              "progress_kpis": [
                {
                  "id": "kpiId",
                  "name": "work completed",
                  "value": "0.3"
                },
                {
                  "id": "kpiId2",
                  "name": "hours spent vs hours planned",
                  "value": "0.8"
                }
              ],
              "id": 2,
              "name": "Install Material Lift or Crane",
              "description": "obj.58195| task for Install Material Lift or Crane",
              "execution_time": {
                "planned": {
                  "id": 1,
                  "workorder_result_id": 2,
                  "logattributes_id": 1,
                  "value": 0,
                  "comments": null,
                  "start": "2020-06-01 14:13:00",
                  "end": "2020-06-05 14:13:00",
                  "real": {
                    "id": 1005,
                    "workorder_result_id": null,
                    "logattributes_id": 1,
                    "value": 1,
                    "comments": "mockup",
                    "start": "2020-06-01 14:13:00",
                    "end": "2020-06-04 14:13:00"
                  }
                },
                "costs": {
                  "planned": {
                    "id": 2,
                    "workorder_result_id": 2,
                    "logattributes_id": 2,
                    "value": 0,
                    "comments": null
                  },
                  "real": {
                    "id": 1006,
                    "workorder_result_id": null,
                    "logattributes_id": 2,
                    "value": 2,
                    "comments": "mockup"
                  }
                },
                "locations": [
                  {
                    "id": 4,
                    "location_parent_id": 3,
                    "name": "NW corner",
                    "building_storey_id": "h29i28hn34gwfd",
                    "building_zone_id": "379930dfdefefewy4456",
                    "building_space_id": "fbeui09jw9fbowigohip"
                  },
                  {
                    "type": "noise",
                    "notes": "Please be advised that during this task, electricity will be cut in this area"
                  },
                  {
                    "type": "noise",
                    "notes": "Please be advised that during this task, noise levels will be higher than normal"
                  }
                ],
                "building_components": [
                  {
                    "id": "sdgskgmfsdg545ldfd",
                    "type": "Wall",
                    "name": "Wall13484"
                  },
                  {
                    "id": "dknfdk49h301525hfjhd",
                    "type": "DoorFrame",
                    "name": "DoorFrame36"
                  }
                ],
                "progress_kpis": [
                  {
                    "id": "kpiId",
                    "name": "work completed",
                    "value": "0.3"
                  },
                  {
                    "id": 3,
                    "name": "Install Safety Measure",
                    "description": "obj.58018| task for Install Safety Measure",
                    "execution_time": {
                      "planned": {
                        "id": 4,
                        "workorder_result_id": 3,
                        "logattributes_id": 1,
                        "value": 0,
                        "comments": null,
                        "start": "2020-06-08 14:13:00",
                        "end": "2020-06-10 14:13:00",
                        "real": {
                          "id": 53,
                          "workorder_result_id": 3,
                          "logattributes_id": 1,
                          "value": 1155,
                          "comments": null,
                          "start": "2020-06-05 14:13:00",
                          "end": "2020-06-09 14:13:00"
                        }
                      },
                      "costs": {
                        "planned": {
                          "id": 5,
                          "workorder_result_id": 3,
                          "logattributes_id": 2,
                          "value": 0,
                          "comments": null
                        },
                        "real": {
                          "id": 1009,
                          "workorder_result_id": null,
                          "logattributes_id": 2,
                          "value": 0,
                          "comments": "mockup"
                        }
                      },
                      "locations": [
                        {
                          "id": 4,
                          "location_parent_id": 3,
                          "name": "NW corner",
                          "building_storey_id": "h29i28hn34gwfd",
                          "building_zone_id": "379930dfdefefewy4456",
                          "building_space_id": "fbeui09jw9fbowigohip"
                        },
                        {
                          "type": "noise",
                          "notes": "Please be advised that during this task, electricity will be cut in this area"
                        },
                        {
                          "type": "noise",
                          "notes": "Please be advised that during this task, noise levels will be higher than normal"
                        }
                      ],
                      "building_components": [
                        {
                          "id": "sdgskgmfsdg545ldfd",
                          "type": "Wall",
                          "name": "Wall13484"
                        },
                        {
                          "id": "dknfdk49h301525hfjhd",
                          "type": "DoorFrame",
                          "name": "DoorFrame36"
                        }
                      ],
                      "progress_kpis": [
                        {
                          "id": "kpiId",
                          "name": "work completed",
                          "value": "0.3"
                        }
                      ]
                    }
                  }
                ]
              }
            }
          ]
        }
      }
    }
  }
}
```

Figure 27 Retrieved JSON file from BIF 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 28 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 29 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 a comment about a specific task. This comment could include text, photographs, videos or even audio recordings. Each time when the user picks/takes a photo, video or audio recording the system stores them locally with a unique and specific name in order to retrieve them from the right directory path and send the new issue (with the attachments) to the BIF through the BIF API.

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. The 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, the system takes his/her 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 system follows the same procedure as the procedure that has been followed for the task comments on the “Task Details” screen.

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 send comments to the related tasks by tapping on them, and navigating to the “task details” screen of the related task when he/she can send photographs, videos and text or a 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 system takes the notifications from a JSON file and displays them into the user's Notification screens. The system filters the notifications from the JSON file in order to show only the ones that are related to the user. The user can reply to a specific notification by commenting about the related task. The system navigates the user to the "Task Details" screen when he/she taps on a notification.

The user will be able to manage his/her notifications and turn them on/off from the Settings screen. The system stores this options' settings to a JSON file and shows up on the notifications based on these options.

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 or more general Info by tapping on menu items:

- "My Issues": The system stores every new submitted issue locally and displays them on the My Issues screen as a list. The user is able to see/listen to the submitted attachments.
- "My Account" (name, location info, mail): After successful authentication the system stores in a JSON file the user information and displays it on the My Account Screen.
- "About BIMERR": The user can be informed about PWMA for Residents application and the BIMERR project on this screen. Additionally he/she can tap on a link which navigates him/her to the BIMERR webpage.
- "FAQs & Tips".
- "Privacy Policy".
- "Log Out".

The figure 30 shows the application workflow which is described in this section.

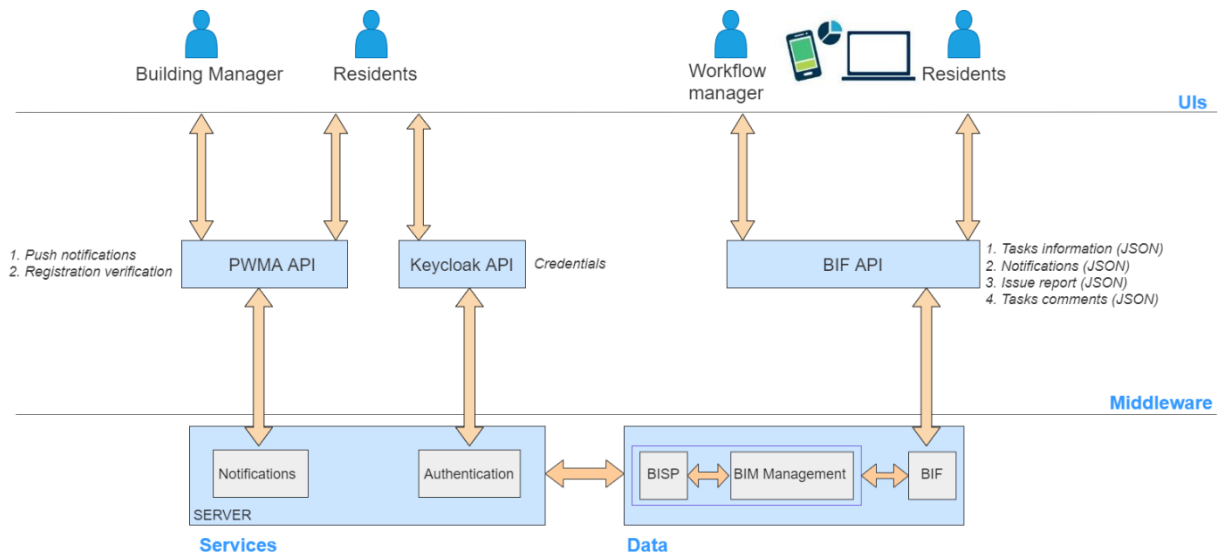


Figure 30 Application workflow

6 INTEGRATION WITH THE OTHER BIMERR COMPONENTS

The PWMA for Residents application has to integrate with the other BIMERR components in order to send and receive data. This integration needs API calls with BIF and PWMA backend.

The BIMERR Interoperability Framework is used to facilitate the various data collection and exchanges from and between the set of the tools and services in the BIMERR ecosystem. These data exchanges are based on the concept of the Representational State Transfer.

6.1 TASK DATA & NOTIFICATIONS

For the projection of all the related tasks with the user, the system sends a request (using UnityWebRequest¹³ (Figure 32)) to the BIF using an endpoint. BIF has already includes a data collection job about annotations (tasks, notifications) which offers an endpoint for API calls (Figure 31). BIF sends, as a reply to the system's request, the data (annotations data model) about tasks in a JSON format and the system stores it in a JSON file format. The system provides a JSON parser for this data annotation model for being able to use it. The system filters which tasks are related to the user based on his/her location and displays them on the "Task List" screen. This file includes the notifications too. In the same way the system displays the related notifications to the "Notifications" screen.



Figure 31 BIF upload to API endpoint

¹³ <https://docs.unity3d.com/ScriptReference/Networking.UnityWebRequest.html>

```

using UnityEngine;
using UnityEngine.Networking;
using System.Collections;

@ Unity Script | 0 references | 0 changes | 0 authors, 0 changes
public class MyBehavior : MonoBehaviour
{
    @ Unity Message | 0 references | 0 changes | 0 authors, 0 changes
    void Start()
    {
        StartCoroutine(Upload());
    }

    1 reference | 0 changes | 0 authors, 0 changes
    IEnumerator Upload()
    {
        WWWForm form = new WWWForm();
        form.AddField("myField", "myData");

        using (UnityWebRequest www = UnityWebRequest.Post("http://www.my-server.com/myform", form))
        {
            yield return www.SendWebRequest();

            if (www.result != UnityWebRequest.Result.Success)
            {
                Debug.Log(www.error);
            }
            else
            {
                Debug.Log("Form upload complete!");
            }
        }
    }
}

```

Figure 32 UnityWebRequest Example Code

6.2 PUSH NOTIFICATIONS & REGISTRATION VERIFICATION

The system uses the notification server of PWMA and BIF to exchange data regarding the registration verification and the push notifications correspondingly.

For push notifications:

As it is mentioned above in section 6.1 for receiving the notifications the system has to send a request (using UnityWebRequest) to the BIF using an endpoint. BIF has already includes a data collection job about annotations (tasks, notifications) which offers an endpoint for API calls. BIF sends, as a reply to the system's request, the annotations data model in a JSON format and the system stores it in a JSON file format. After that the system checks if there is any H&S notification and task change or any tasks that takes place in that day. If so it uses the Unity Mobile Notifications package¹⁴ to create android notifications to display them on user's device as push notifications (Figure 33).

¹⁴ <https://docs.unity3d.com/Packages/com.unity.mobile.notifications@1.4/manual/index.html>

```

if (((int.Parse(realStartDate[0]) == DateTime.UtcNow.Year) && (int.Parse(realStartDate[1]) <=
DateTime.UtcNow.Month) && (int.Parse(realStartDate[2]) <= DateTime.UtcNow.Day)) && ((int.Parse
(realEndDate[0]) == DateTime.UtcNow.Year) && (int.Parse(realEndDate[1]) >= DateTime.UtcNow.Month) &&
(int.Parse(realEndDate[2]) >= DateTime.UtcNow.Day)))
{
    var channel = new AndroidNotificationChannel()
    {
        Id = "channel_id",
        Name = "Notifications Channel",
        Importance = Importance.Default,
        Description = "Reminder notifications",
    };
    AndroidNotificationCenter.RegisterNotificationChannel(channel);
    //create the notifications that is going to be sent
    var notification = new AndroidNotification();
    notification.Title = "Reminder";
    notification.Text = (char)34 + t1.name + +(char)34 + " task is taking place today";
    notification.FireTime = System.DateTime.Now.AddSeconds(15);
    //send the notification
    var id = AndroidNotificationCenter.SendNotification(notification, "channel_id");
    // if the script is run and a message is already, cancel it and re-schedule another message
    if (AndroidNotificationCenter.CheckScheduledNotificationStatus(id) ==
        NotificationStatus.Scheduled)
    {
        AndroidNotificationCenter.CancelAllNotifications();
        AndroidNotificationCenter.SendNotification(notification, "channel_id");
    }
}
}

```

Figure 33 Android Notifications Example

For the registration verification:

The system has to send the data of the resident, who wants to register, to the building manager for confirmation. If the building manager approves the registration, system has to send back to the resident a unique code for being able to finish his/her registration. To make this possible, system uses the notification server of PWMA. More specifically the system sends the resident's data through an API channel to the building manager and the system (from the building manager's side) receives it and displays it as a push notification. If the building manager approves the resident's request, the system produces a unique code and sends it to the user through another API channel and displays it to the user as a push notification.

6.3 ISSUE REPORTING & TASKS COMMENTS

For the issue reporting and tasks comments procedure the system has to send every new issue to the BIF. To make this possible, a new collection job has to be created on the BIF platform. In the first stage of the collection setup, the name and the pre-processing attributes of the job are defined (Figure 34).

The screenshot shows a web form titled 'Job Details' with the subtitle 'Basic information about the job'. It contains two main sections: 'NAME' and 'DESCRIPTION'. The 'NAME' field is a text input containing 'Annotation Data Mock'. The 'DESCRIPTION' field is a larger text area with the placeholder text 'Enter a short description for your job'. Below these fields, there is a 'Pre-processing' section with the subtitle 'Before importing your data to the platform, you can use additional tools to better prepare them'. This section contains three checked checkboxes: 'HARVESTER' (Collect data through files, external APIs or other services), 'MAPPING' (Map your data to the common data model), and 'LOADER' (Load the processed data to the data storage).

Figure 34 Annotation Data collection setup 1

In the second stage of the setup, the data loading needs to use the **Platform's API** option so the direct uploading to BIF's internals is possible (Figure 35).

The screenshot shows a web form titled 'Configure Harvester: Annotation Data Mock'. It has a progress bar at the top with two steps: 'STEP 1: Setup Harvester Service' and 'STEP 2: Test and Harvester Configuration'. The current step is 'STEP 2'. The form is titled 'Data Loading' with the subtitle 'How do you plan to load your data to the platform?'. It contains three radio button options: 'FILE UPLOAD' (Direct file upload (CSV, JSON, XML)), 'DATA PROVIDER'S AVAILABLE API' (Collect data from the APIs provided by applications and systems of the data provider or from open APIs), and 'PLATFORM'S API' (Upload data to the Platform's APIs). The 'PLATFORM'S API' option is selected.

Figure 35 Annotation Data collection 2

In the third stage of the setup, the data type and processing time are selected while a sample of the data to be uploaded is provided (Figure 36).

The form is titled 'Annotation Data collection setup 3' and contains four sections:

- Data Loading:** 'How do you plan to load your data to the platform?' with a radio button selected for 'PLATFORM'S API' (Upload data to the Platform's APIs).
- Data Type:** 'What type of data do you intend to load?' with radio buttons for 'Text only' (selected) and 'Text and Binary'.
- Processing:** 'How often should we process your data?' with radio buttons for 'Immediately' (selected), 'On an hourly basis', 'On a daily basis', and 'On a weekly basis'.
- Sample Upload:** 'Upload a sample of your data to be used in next steps'. It includes a 'BROWSE' button and a text field showing 'annotations.json' and '570.0 B'.

Figure 36 Annotation Data collection setup 3

Finally, in the last stage of the collection setup the endpoint that will be used in the Annotation Data send request is created. In addition an overview of the uploaded data is provided (Figure 37).

The form is titled 'Annotation Data collection setup 4' and contains three main sections:

- Method & URL:** 'The API method and the full URL'. It shows 'POST' and the URL 'https://bimerr.s5labs.eu/api/upload/07f3afc6-3b25-43b0-a63a-c39a1e7faf41'.
- Instructions:** 'How to use the POST endpoint'. It includes an 'Authentication' section with instructions: 'In order to use the generated API, you should be authenticated. To do so, you need to: 1. Use an already generated access token with upload scope or generate a new one. This token will be used to authenticate your requests. 2. Add the created access token into an X-API-TOKEN header in your request. 3. Insert the data you wish to upload to the API, into the body of your request, in JSON format.'
- Data Sample:** 'The details of the data sample that was uploaded'. It shows a JSON snippet:

```
{  "Topic": {    "definition": "Topic contains reference information of the topic."  }}
```

Figure 37 Annotation Data collection setup 4

Once an Annotation is created based on user input from the appropriate action(issue reporting, task comments), the locally parsed and agreed Annotation Data model is serialized and send as a REST POST request to the endpoint URL that was provided by BIF previously (Figure 38).

```
[Serializable]
public class HasTopic
{
    public string Identifier;
    public string TopicType;
    public string TopicStatus;
    public object Title;
    public object Label;
    public string CreationDate;
    public string CreationAuthor;
    public string AssignedTo;
    public object Description;
    public string Stage;
    public string TaskId;
    public RelatedFile RelatedImageFile;
    public RelatedFile RelatedVideoFile;
    public RelatedFile RelatedAudioFile;
    public Related RelatedComponent;
    public Related RelatedSpace;
    public RelatedFile RelatedIfcFile;
    public HasComment HasComment;
}

[Serializable]
public class HasComment
{
    public string Identifier;
    public string CreationDate;
    public string CreationAuthor;
    public string Comment;
}
```

Figure 38 Annotation data model example

When the user reports an issue or comments on a task, he/she is able to attach media files (photos, videos, audio recordings). The system for being able to upload this type of data to BIF has to send them one by one as information objects. The information object includes the media file in binary format and an id of it. From the other side the annotations data model includes an array with the attachments' id of each related attachments.

7 CONCLUSION AND OUTLOOK

This deliverable introduces the second and updated 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 whole process of PWMA for residents application. It provides details on how the design process was laid out. Furthermore the use cases and the user's requirements which are based on deliverable 3.1, are analysed. Subsequently, it describes the methodology which was followed and the decision of the software support for the implementation of the application. Furthermore, it analyses the implementation and the features of application as well as the workflow of it. Afterwards, it gives a description of how the application integrates with the other BIMERR components. Finally it provides a detailed guideline for the users on how to use the app.

Overall, the final version of the PWMA for residents application, delivered in M30 of the project's implementation is built upon the feedback received from partners on its initial version and a limited test run during the pre-validation activities at KRIPIS.

8 BIBLIOGRAPHY

- [1] BIMERR, "DELIVERABLE D3.1 Stakeholder requirements for the BIMERR system," 2019.
- [2] Chen, J.W. & Zhang, J., "Comparing Text- -based and Graphic User Interfaces for novice and expert users," in *AMIA...Annuals Symposium proceedings. AMIA Symposium*, 2007.
- [3] Chadwick-Dias, A., McNulty, M. and Tullis, T, "Web usability and age: how design changes can improve performance," in *In Proceedings of the 2003 conference on Universal usability. Association for Computing Machinery*, New York, NY, USA, 2002.
- [4] Kascak, L. R., Rébola, C. B., and Sanford, J. A., "Integrating Universal Design (UD) Principles and Mobile Design Guidelines to Improve Design of Mobile Health Applications for Older Adults," in *IEEE International Conference on Healthcare Informatics*, Verona, 2014.
- [5] N. B. Ruparelia, *Software development lifecycle models*, SIGSOFT Softw, 2010.
- [6] Wynn, D.C., Clarkson, P.J., "Process models in design and development," pp. 161-202, 2018.
- [7] Larman, C. and Basili, V. R., "Iterative and Incremental Development: A Brief History," pp. 47-56, June 2003.
- [8] 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., "Principles behind the Agile Manifesto," Agile Alliance. Archived from the original, 2010.

ANNEX I – USER MANUAL

The user manual contains all the necessary information for the user to make full use of the information system. This manual includes descriptions of system functions and capabilities, emergencies and alternative operating modes, and step-by-step procedures for accessing and using the system.

DESCRIPTION ON HOW TO USE/OPERATE

To use PWMA for Residents application (on an Android device):

1. Download the Application from the BIMERR official webpage (Android 8+ is required)
2. Installation:
 - Open the file browser application and then the download folder. Find the PWMAForResidents.apk and tap on. You will see this message: “For your security, your phone is not allowed to install unknown apps from this source. “ (Figure 39).

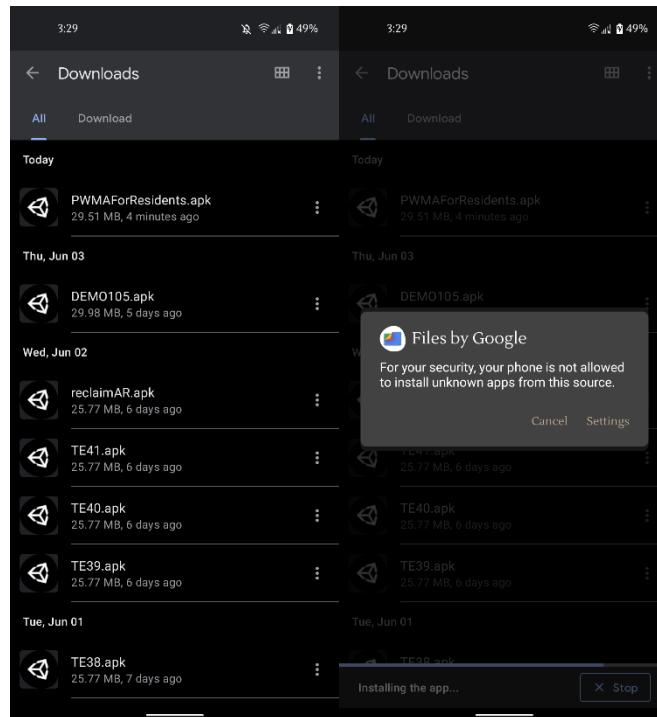


Figure 39 Installation Procedure 1

- Tap on settings.
- Toggle on “Allow from this source” and go back (Figure 40).

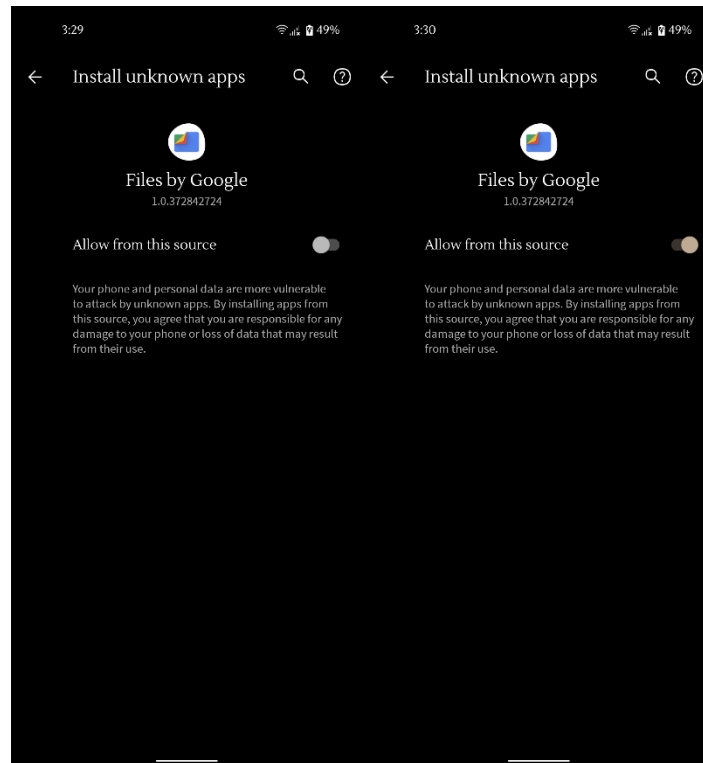


Figure 40 Installation Procedure 2

- Tap on install (Figure 41).
- Tap on install anyway (Figure 41).

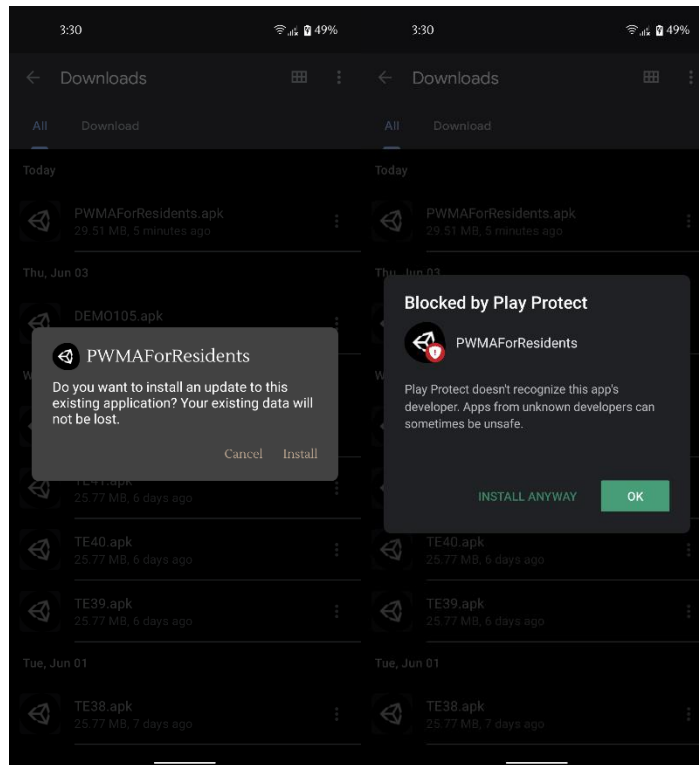


Figure 41 Installation Procedure 3

- The app should safely be installed.
3. Open the application and register by entering your full name, your email, the password of your choice, your number, your building and apartment that you live in (Figure 42).

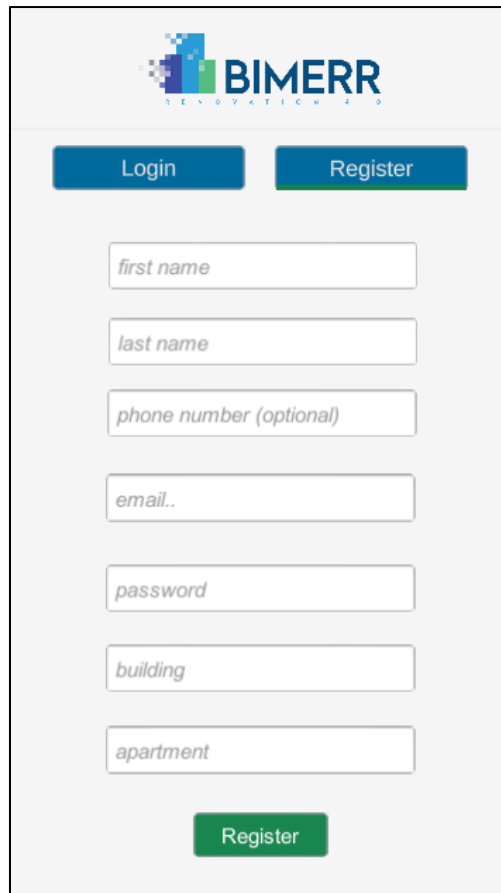
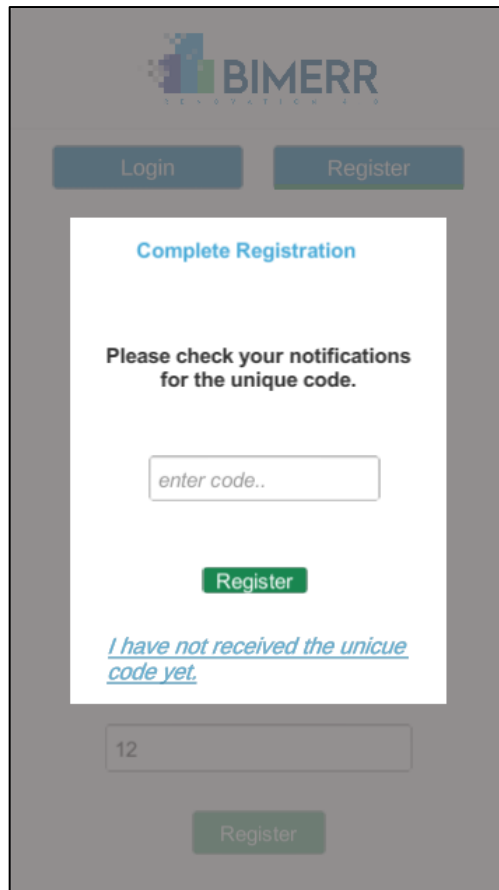
The image shows a mobile application interface for the BIMERR project. At the top, there is a logo with a stylized 'B' made of blue and green squares, followed by the text 'BIMERR' and 'TECHNOLOGY' in smaller letters below it. Below the logo, there are two buttons: 'Login' in a blue box and 'Register' in a green box. Under these buttons, there are seven input fields stacked vertically, each with a placeholder text: 'first name', 'last name', 'phone number (optional)', 'email..', 'password', 'building', and 'apartment'. At the bottom of the form, there is a green 'Register' button.

Figure 42 Register Screen

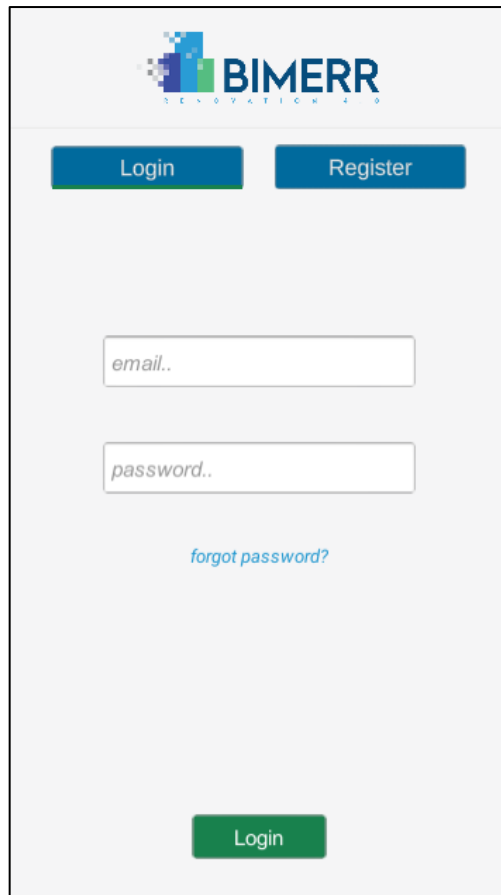
4. Wait for the Building manager's authentication/confirmation.
5. You will receive a unique code as a push notification. Fill it in at the pop up menu which is displayed (Figure 43).
6. If you will not receive it between half an hour, tap on "I have not received the unique code yet" (Figure 43).



The image shows a mobile application interface for BIMERR. At the top, there is a logo with the text "BIMERR" and "REGISTRATION" below it. Below the logo are two buttons: "Login" and "Register". The main content area is a white box with the title "Complete Registration" in blue. Below the title, it says "Please check your notifications for the unique code." There is a text input field with the placeholder "enter code..". Below the input field is a green "Register" button. At the bottom of the white box, there is a link that says "I have not received the unique code yet." Below the white box, there is a text input field with the number "12" and a green "Register" button.

Figure 43 Registration verification

7. Once you have completed the steps above, you will be directed to the “Login” screen (Figure 44).
8. Login entering your email and your password.



The image shows a login screen for the BIMERR system. At the top, there is a logo for BIMERR with the tagline 'INNOVATION IN REPAIRS'. Below the logo, there are two buttons: 'Login' and 'Register'. In the center, there are two input fields: one for 'email..' and one for 'password..'. Below these fields, there is a link that says 'forgot password?'. At the bottom, there is a green 'Login' button.

Figure 44 Login Screen

9. Once your credentials are approved, you will be directed to the “Task List” screen (Figure 45).
10. The “Task List” screen displays all the tasks of the workorders which are related to you according to your apartment’s location. In each task there is a bar which shows the progress of it, the type of the Health & Safety issues that may come up, the progress percentage of the task, and flags which represent when the task has to start/end (blue flags) and when actually the task starts/ends (green flags) (Figure 46)
 - Blue colour corresponds to the planned date of the task
 - Green colour corresponds to the real starting date.
 - Orange colour corresponds to the delayed time based on planned date and the real date.

At the top of the screen you can select to see only the completed/pending/ongoing tasks.

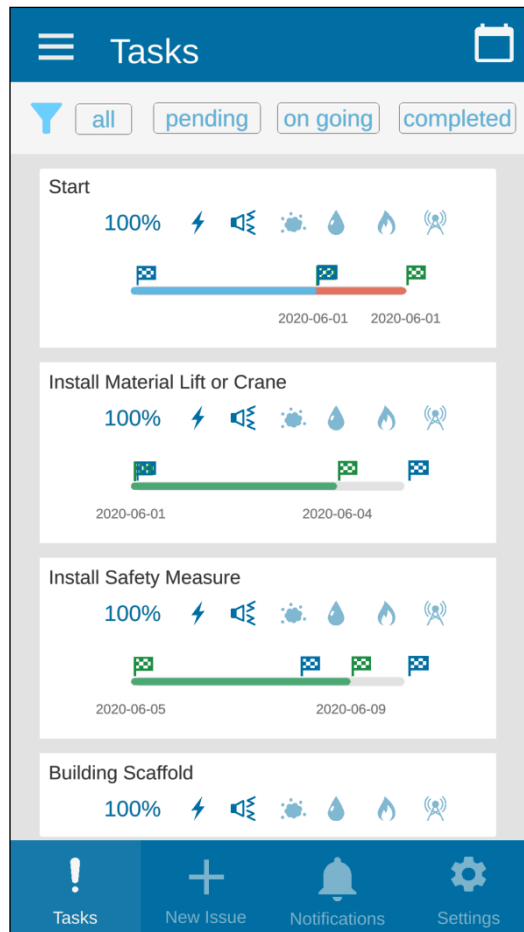


Figure 45 Tasks List Screen

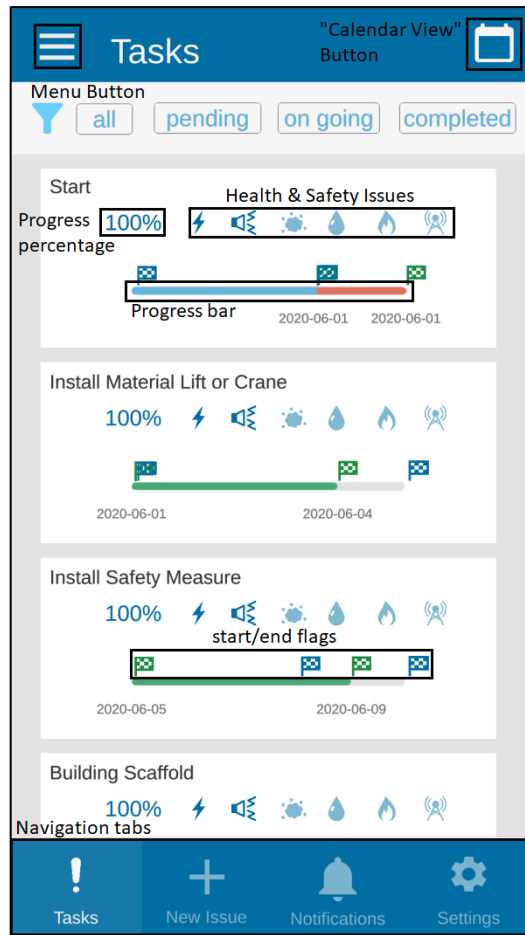


Figure 46 Tasks List Screen (2)

11. You can see more details of each task by tapping on them. You will be directed to the “Task Details” screen where you can see additional information such as the exact duration of the task, the date range of the task, task ID and the location of the task (Figure 47). You have the capability to comment the task writing a text taking a photograph or a video or even making an audio recording. If you want to send an audio recording tap and hold the mic icon. The device starts to record since you tap the mic and stops recording when you stop holding it. The audio file will be attached automatically. If you want to capture an image/ video tap on camera/ video icon and tap the capture button (Figure 48, 49). Now you have to pick the media file which you have captured from your gallery to attach to the comment. You are able to pick any image or video that you have on your phone gallery too (Figure 50).

< Start

A horizontal progress bar with three segments. The first segment (from start to 2020-02-01) is blue with a flag icon at the end. The second segment (from 2020-02-01 to another 2020-02-01) is red with a flag icon at the end. The third segment (after the second 2020-02-01) is green with a flag icon at the end.

Start: delayed Finish: delayed Issues: 1

100%

Time: 08:00:00 - 08:10:00

Date Range:
2020-02-01 - 2020-02-01

Locations:

Location1: building, apartment
 Location2: building, apartment

Task ID: 1

Comment: Send

Add a comment...

Figure 47 Task Details Screen

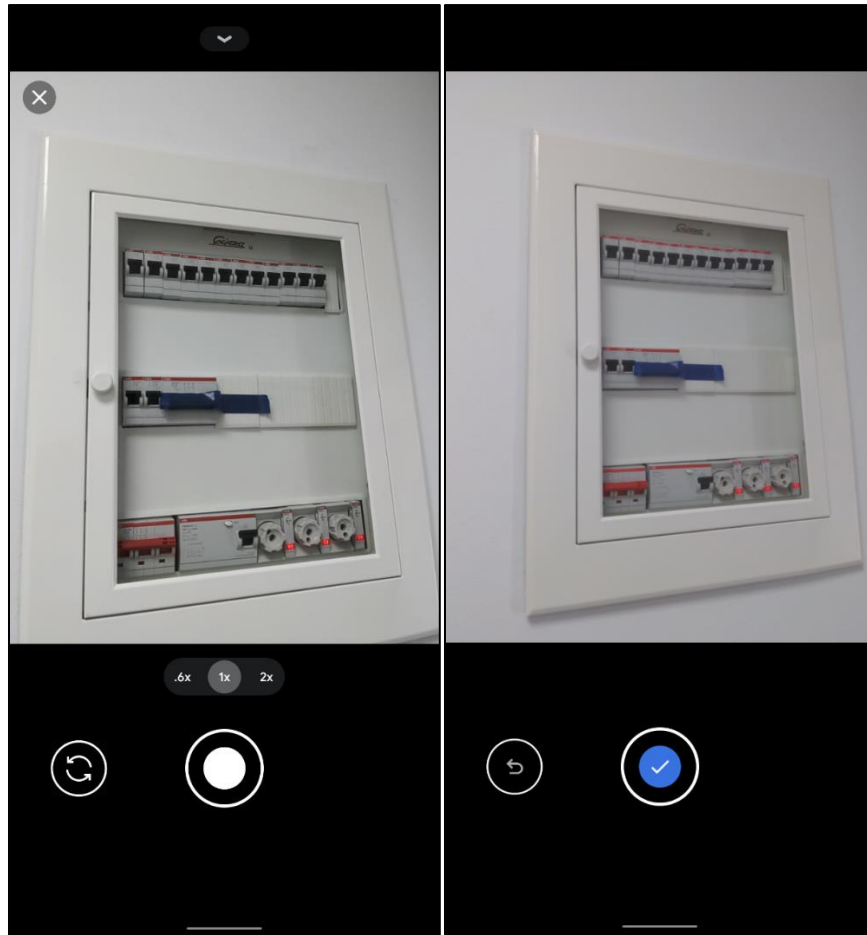


Figure 48 Camera Capture/Confirmation



Figure 49 Video Capture/Confirmation



Figure 50 Pick from Gallery

12. Back to the Task List screen there is calendar icon at the top left corner which navigates you to the “Calendar View” screen (Figure 46, 51). There you can see all the tasks in a calendar view mode and you can see additional details for each of them by tapping on it, at the bottom of the screen (Figure 52).

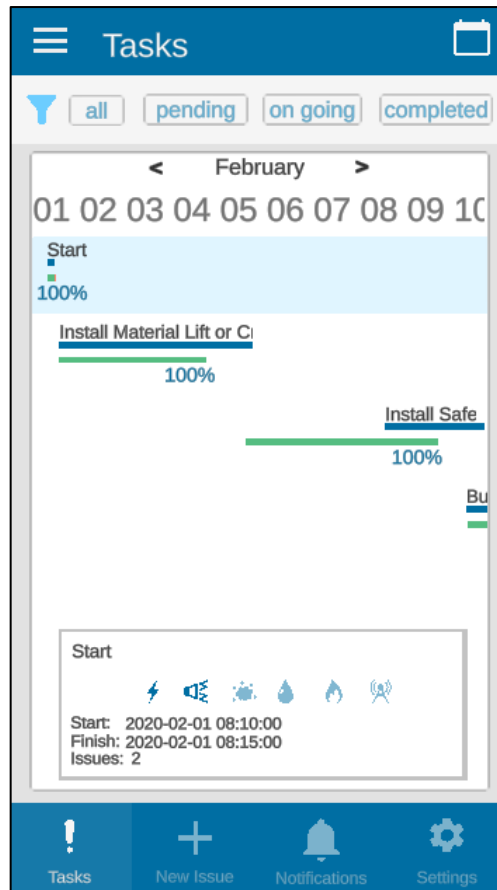


Figure 51 Calendar View Screen

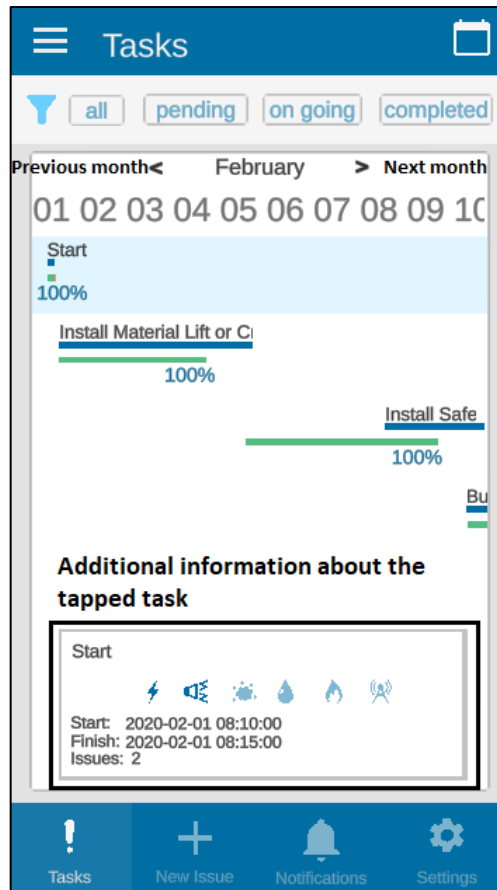
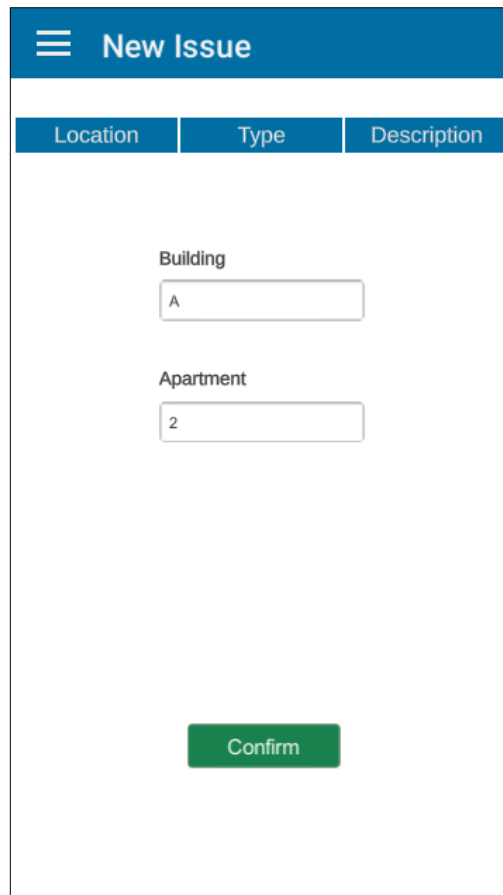


Figure 52 Calendar View Screen 2

13. At the bottom of the “Task List” screen there are four tabs (Figure 46) where you can select to navigate to:

-“New Issue” Screen. You select this screen if you want to submit an issue that you have noticed in your apartment. New issue submission includes 3 different screens.

- On the first screen you have to fill in your location (Figure 53)



The image shows a mobile application screen titled "New Issue". At the top, there is a blue header bar with a white hamburger menu icon on the left and the text "New Issue" in white. Below the header, there is a table with three columns: "Location", "Type", and "Description". The table is currently empty. Below the table, there are two text input fields. The first field is labeled "Building" and contains the letter "A". The second field is labeled "Apartment" and contains the number "2". At the bottom of the screen, there is a green button with the text "Confirm" in white.

Location	Type	Description
----------	------	-------------

Building
A

Apartment
2

Confirm

Figure 53 New Issue Screen 1

- On the second screen you have to select the type of issue that you have encountered (Figure 54)

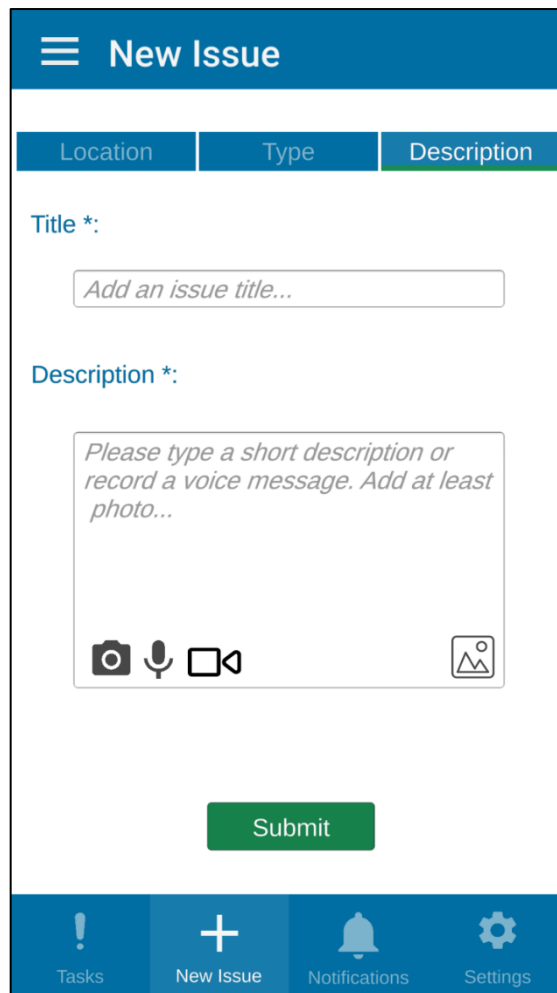
Location	Type	Description
	Health & Safety (danger, safety, security, inconvenience)	<input type="radio"/>
	Malfunction - Repair (does not work properly or at all)	<input type="radio"/>
	Maintenance (works but maintenance is needed)	<input type="radio"/>
	Functional Suggestion (not easy to use, affords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

Next

Tasks New Issue Notifications Settings

Figure 54 New Issue Screen 2

- On the third screen you have to put a title and a description to the issue. You have also the capability to add attachments on the issue by writing a text, taking a photograph or a video or even making an audio recording. This procedure has already been described in 11th step.
Finally to send the issue to the building manager you have to press the “submit” button (Figure 55).



The image shows a mobile application screen titled "New Issue". At the top, there is a blue header bar with a hamburger menu icon and the text "New Issue". Below the header, there is a table with three columns: "Location", "Type", and "Description". The "Description" column is highlighted with a green underline. Below the table, there is a form with two main sections. The first section is labeled "Title *:" and contains a text input field with the placeholder text "Add an issue title...". The second section is labeled "Description *:" and contains a larger text area with the placeholder text "Please type a short description or record a voice message. Add at least photo...". Below the text area, there are four icons: a camera, a microphone, a video camera, and a photo gallery icon. At the bottom of the form, there is a green "Submit" button. At the very bottom of the screen, there is a blue navigation bar with four icons: an exclamation mark (Tasks), a plus sign (New Issue), a bell (Notifications), and a gear (Settings). The "New Issue" icon is highlighted.

Figure 55 New Issue Screen 3

-Notifications Screen. You have tap this tab if you want to be informed about health & safety notifications or task changes. The notifications are displayed as a list and you can scroll up and down on them. You will also be informed by android notifications(Figure 56). If you want to comment on any of them you have to tap on it and you will be directed to the “Task Details” screen (Figure 57).

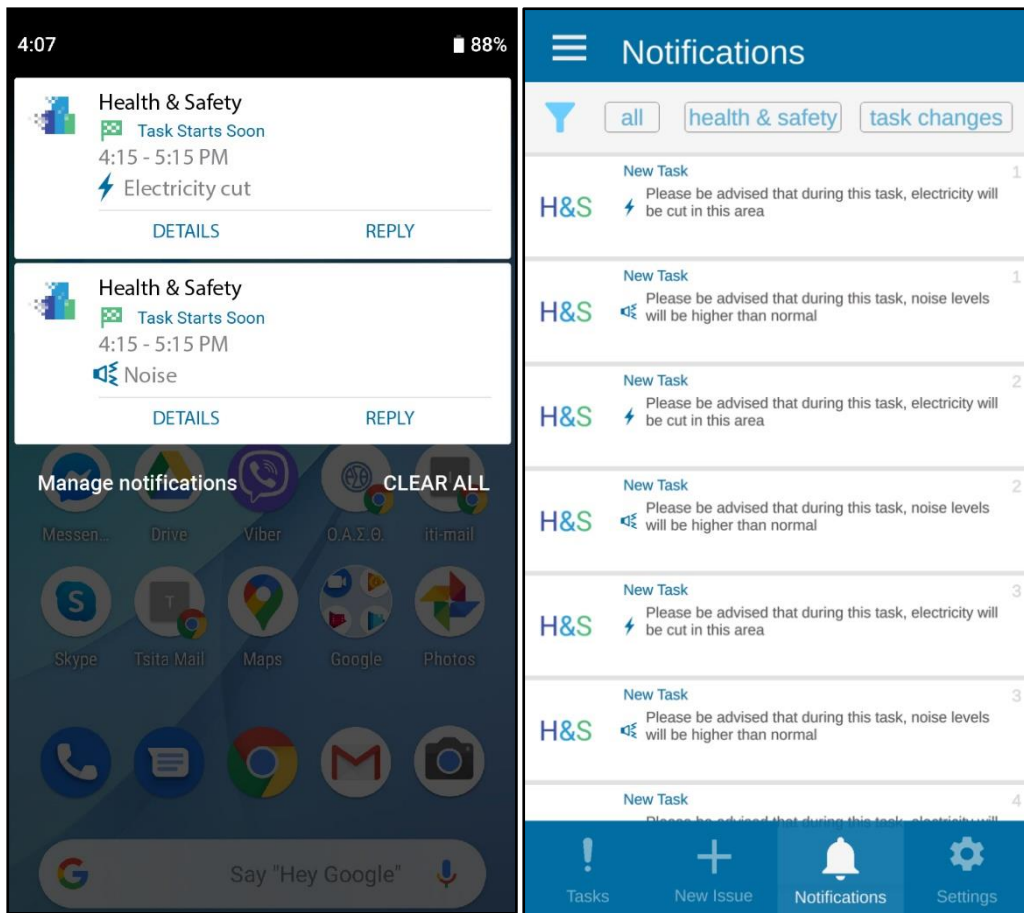


Figure 56 Android Notifications (Left), In App Notifications (Right)

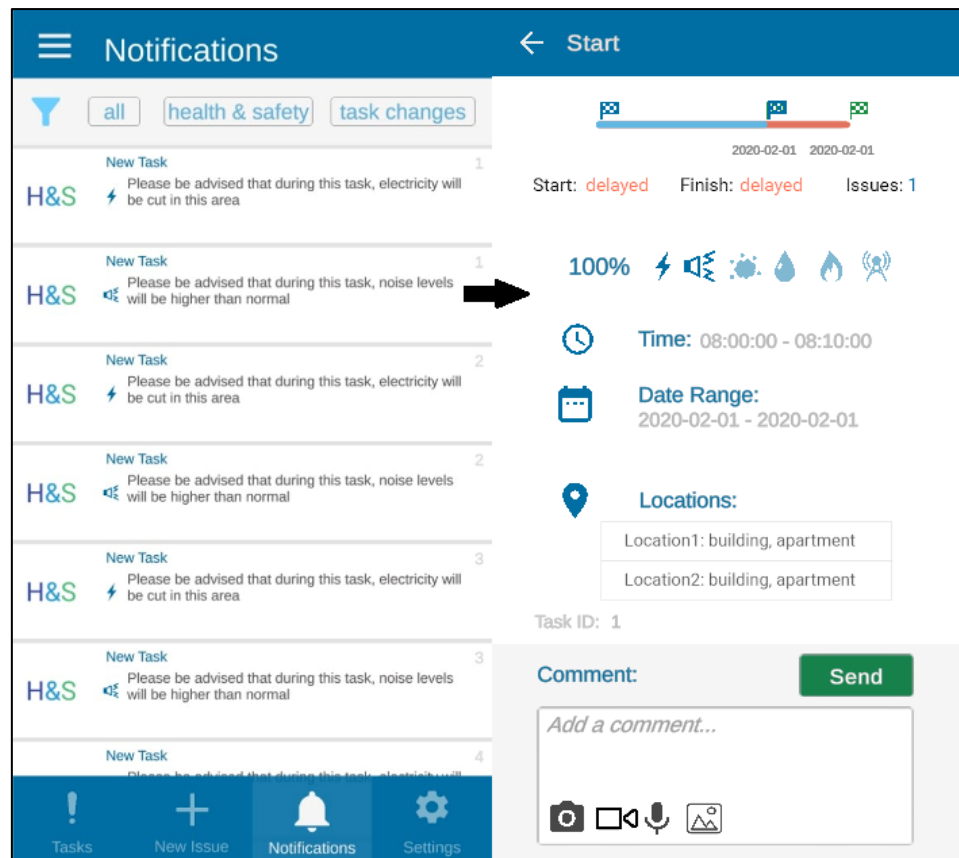


Figure 57 Comment a Notification

-Notifications Settings. You have to tap on this tab if you want to customize your notifications by turning them on/off (Figure 58).

- 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.
 - 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.

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.

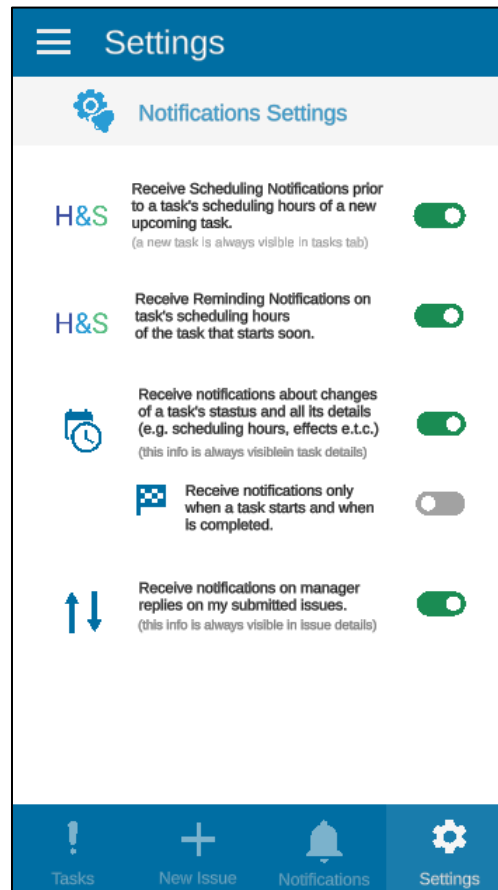


Figure 58 Notifications Settings Screen

14. When you are on any of these four screens you can always tap on the burger menu icon at the top left of your screen and you will be directed to the menu (Figure 59). On the menu screen you can select any of the following menu items:

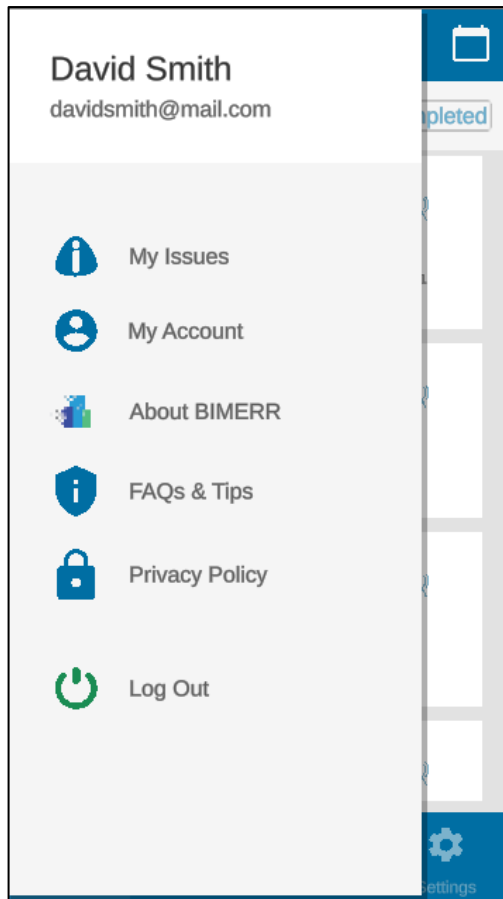


Figure 59 Menu Screen

- My Issues, if you want to see your submitted issues. You are able to see every attachment that you have submitted. Tap on the video thumbnail if you want the submitted video to be played. Do the same procedure for the audio recordings (Figure 60).

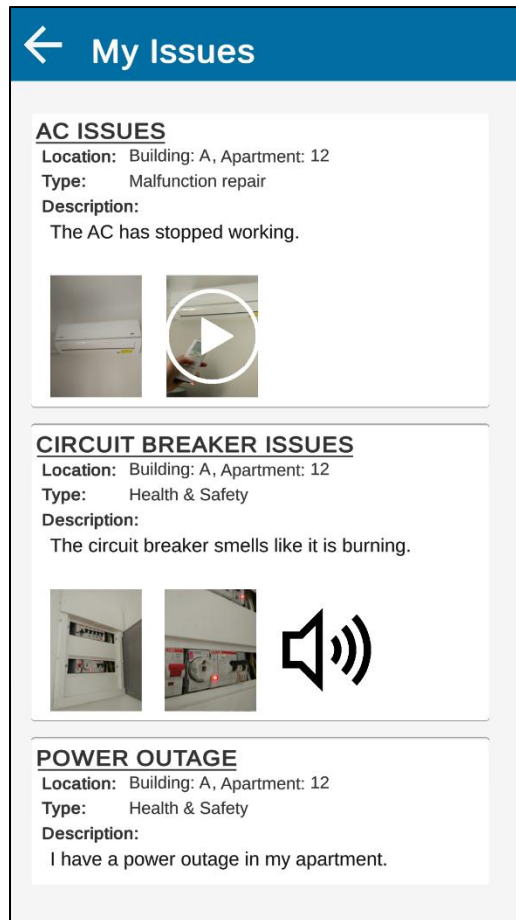


Figure 60 My Issues Screen

- About BIMERR, if you want to be informed about PWMA for Residents application and for the BIMERR project in general. Tap on the link if you want to navigate to the BIMERR webpage (Figure 61).

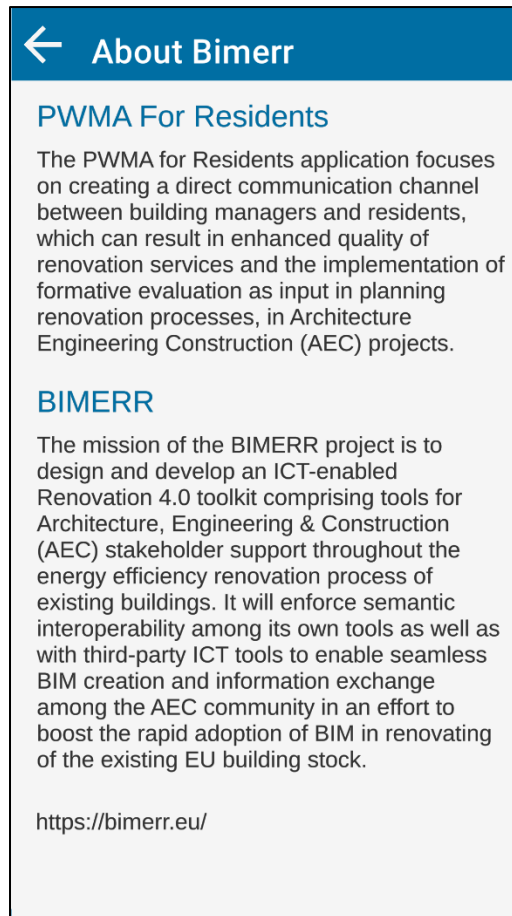


Figure 61 About BIMERR

- My Account, if you want to check your personal info.
- Privacy Policy, if you want to be informed about it.
- Log out, if you want to log out from the application (Figure 62).

You can always exit from the menu by tapping on the outside area of the menu.

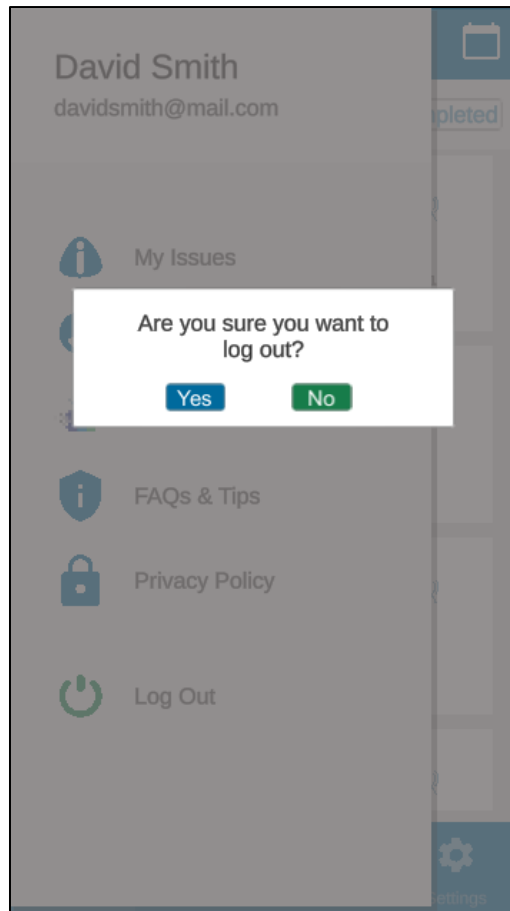


Figure 62 Log Out

TROUBLESHOOTING

Follow the next steps, if you encounter any issue while using the app.

If there is a long delay while waiting for receiving the unique code, as you try to register, please restart the app and be patient for the building manager to authenticate your registration data.

If you observe that there is no list of tasks in the first use of the application please restart the application.

If there is any lag while using the application, please restart it.

If you don't receive new notifications and push notifications, please check your internet connection.

If some of the previous issues don't be fixed, please check if there is an application's update available