DELIVERABLE D3.4
Collaborative Platform for Work Process Support

Deliverable Status: Final
File Name: SatisFactory-D3.4-v1.0-Collaborative platform for work process support.pdf
Due Date: August 2016 (M20)
Submission Date: August 2016 (M20)
Task Leader: FIT

<table>
<thead>
<tr>
<th>Dissemination level</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
<td>X</td>
</tr>
<tr>
<td>Confidential, only for members of the Consortium (including the Commission Services)</td>
<td></td>
</tr>
</tbody>
</table>

This project has received funding from the European Union’s Horizon 2020 Research and innovation programme under Grant Agreement n°636302
The SatisFactory project consortium is composed of:

<table>
<thead>
<tr>
<th>Consortium</th>
<th>Name</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>CERTH</td>
<td>Centre for Research and Technology Hellas</td>
<td>Greece</td>
</tr>
<tr>
<td>SIGMA</td>
<td>Sigma Orionis SA</td>
<td>France</td>
</tr>
<tr>
<td>FRAUNHOFER</td>
<td>Fraunhofer-Gesellschaft zur Foerderung der Angewandten Forschung E.V</td>
<td>Germany</td>
</tr>
<tr>
<td>COMAU</td>
<td>Comau SPA</td>
<td>Italy</td>
</tr>
<tr>
<td>EPFL</td>
<td>École Polytechnique Fédérale de Lausanne</td>
<td>Switzerland</td>
</tr>
<tr>
<td>ISMB</td>
<td>Istituto Superiore Mario Boella sulle tecnologie dell’informazione e delle telecomunicazioni</td>
<td>Italy</td>
</tr>
<tr>
<td>ABE</td>
<td>Atlantis Engineering AE</td>
<td>Greece</td>
</tr>
<tr>
<td>REGOLA</td>
<td>Regola srl</td>
<td>Italy</td>
</tr>
<tr>
<td>GlassUP</td>
<td>GlassUp srl</td>
<td>Italy</td>
</tr>
<tr>
<td>QPLAN</td>
<td>Q-PLAN International Advisors LTD</td>
<td>Greece</td>
</tr>
</tbody>
</table>

Disclaimer

This document reflects only the author's views and the European Union is not liable for any use that may be made of the information contained therein.

---

1 Project Coordinator
2 Terminated beneficiary since June 2016 and replaced by QPLAN
### Authors List

<table>
<thead>
<tr>
<th>Surname</th>
<th>First Name</th>
<th>Beneficiary</th>
<th>Contact email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leon Rojas</td>
<td>Sarah</td>
<td>FIT</td>
<td><a href="mailto:sarah.leon.rojas@fit.fraunhofer.de">sarah.leon.rojas@fit.fraunhofer.de</a></td>
</tr>
<tr>
<td>Suleri</td>
<td>Sarah</td>
<td>FIT</td>
<td><a href="mailto:sarah.suleri@fit.fraunhofer.de">sarah.suleri@fit.fraunhofer.de</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Surname</th>
<th>First Name</th>
<th>Beneficiary</th>
<th>Contact email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Della Casa</td>
<td>Emiliano</td>
<td>GlassUp</td>
<td><a href="mailto:emiliano.dellacasa@glassup.net">emiliano.dellacasa@glassup.net</a></td>
</tr>
<tr>
<td>2</td>
<td>Ilioudi</td>
<td>Athina</td>
<td>CERTH</td>
<td><a href="mailto:athina.il@iti.gr">athina.il@iti.gr</a></td>
</tr>
<tr>
<td>3</td>
<td>Jentsch</td>
<td>Marc</td>
<td>FIT</td>
<td><a href="mailto:marc.jentsch@fit.fraunhofer.de">marc.jentsch@fit.fraunhofer.de</a></td>
</tr>
<tr>
<td>4</td>
<td>Quandt</td>
<td>Julian</td>
<td>FIT</td>
<td><a href="mailto:julian.quandt@fit.fraunhofer.de">julian.quandt@fit.fraunhofer.de</a></td>
</tr>
<tr>
<td>5</td>
<td>Tsolakis</td>
<td>Apostolos</td>
<td>CERTH</td>
<td><a href="mailto:tsolakis@iti.gr">tsolakis@iti.gr</a></td>
</tr>
</tbody>
</table>
**Revision Control**

<table>
<thead>
<tr>
<th>Version</th>
<th>Author(s)</th>
<th>Date</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>FIT</td>
<td>July 2016</td>
<td>Initial structure</td>
</tr>
<tr>
<td>0.2</td>
<td>FIT</td>
<td>July 2016</td>
<td>Chapter 1, 2, 3, 5</td>
</tr>
<tr>
<td>0.3</td>
<td>FIT</td>
<td>August 2016</td>
<td>List of Abbreviations</td>
</tr>
<tr>
<td>0.4</td>
<td>FIT</td>
<td>August 2016</td>
<td>Section 4.2.1</td>
</tr>
<tr>
<td>0.5</td>
<td>FIT</td>
<td>August 2016</td>
<td>Sections 3.1.1 and 3.1.2, small corrections</td>
</tr>
<tr>
<td>0.6</td>
<td>FIT</td>
<td>August 2016</td>
<td>Moved some sections</td>
</tr>
</tbody>
</table>
| 0.7     | FIT         | August 2016 | • Executive Summary  
• Conclusion  
• Updated Gamification Framework Architecture figure  
• List of Reviewers  
• Spelling & Grammar corrections  
• Incorporated review changes |
| 0.9     | FIT, CERTH  | August 2016 | Chapter 4 |
| 1.0     | FIT         | August 2016 | Ready for submission to the EC |
### List of Reviewers (in alphabetic order)

<table>
<thead>
<tr>
<th>#</th>
<th>Surname</th>
<th>First Name</th>
<th>Beneficiary</th>
<th>Contact email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conzon</td>
<td>Davide</td>
<td>ISMB</td>
<td><a href="mailto:conzon@ismb.it">conzon@ismb.it</a></td>
</tr>
<tr>
<td>2</td>
<td>Metaxa</td>
<td>Ifigeneia</td>
<td>ABE</td>
<td><a href="mailto:metaxa@abe.gr">metaxa@abe.gr</a></td>
</tr>
</tbody>
</table>
Table of Contents

List of Figures .................................................................................................................. 8
List of Tables ..................................................................................................................... 10
List of Definitions & Abbreviations .................................................................................. 11
Executive Summary .......................................................................................................... 13
1 Introduction .................................................................................................................... 14
2 Implementation of Gamification Framework .................................................................... 15
  2.1 Approach and used technologies ............................................................................... 15
    2.1.1 RESTful web API ............................................................................................... 15
    2.1.2 Node.js .............................................................................................................. 15
    2.1.3 CIDEM .............................................................................................................. 15
  2.2 Implementation .......................................................................................................... 15
    2.2.1 Data model ........................................................................................................ 15
    2.2.2 Architecture ..................................................................................................... 17
    2.2.3 API .................................................................................................................. 17
    2.2.4 Visualizing Team Point with Digital Andon System ........................................... 20
3 Implementation of Suggestions Platform ....................................................................... 22
  3.1 Approach and used technologies ............................................................................... 22
    3.1.1 Node.js .............................................................................................................. 22
    3.1.2 Ionic ................................................................................................................ 22
    3.1.3 AngularJS ....................................................................................................... 22
  3.2 Implementation .......................................................................................................... 22
    3.2.1 Architecture ..................................................................................................... 22
    3.2.2 Activity Diagram ............................................................................................. 25
    3.2.3 Server API ....................................................................................................... 25
    3.2.4 Connection to Gamification Framework ............................................................. 26
  3.3 Deployment ................................................................................................................. 26
  4.1 Approach and used technologies ............................................................................... 28
    4.1.1 Technologies ................................................................................................... 28
    4.1.2 Content ........................................................................................................... 29
  4.2 Implementation .......................................................................................................... 31
    4.2.1 Architecture ..................................................................................................... 31
    4.2.2 Back End ......................................................................................................... 32
    4.2.3 Front End ......................................................................................................... 35
5 Protect your Health ......................................................................................................... 57
  5.1 Approach and used technologies ............................................................................... 57
    5.1.1 Raspberry Pi ..................................................................................................... 57
    5.1.2 Raspbian ......................................................................................................... 57
    5.1.3 Python ............................................................................................................ 57
    5.1.4 Putty ............................................................................................................... 58
  5.2 Implementation .......................................................................................................... 58
    5.2.1 Hardware ......................................................................................................... 58
    5.2.2 Software ......................................................................................................... 60
    5.2.3 Connection to Gamification Framework .......................................................... 61
LIST OF FIGURES

Figure 1: CIDEM Compliant Data Model Schema for the Gamification Framework ..................................... 16
Figure 2: Gamification Framework Architecture ...................................................................................... 17
Figure 3: API Calls from components to Gamification Backend ............................................................. 18
Figure 4: Activity Diagram: Visualizing team points with Digital Andon System ...................................... 20
Figure 5: Suggestions platform, component diagram .................................................................................. 23
Figure 6: First version of the suggestion platform data model ................................................................. 24
Figure 7: Second version of the suggestion platform data model ............................................................. 24
Figure 8: Activity diagram suggestions, first part ......................................................................................... 25
Figure 9: Activity diagram suggestions, second part .................................................................................... 25
Figure 10: Deployment diagram suggestions for improvement platform .................................................... 27
Figure 11: SatisFactory Platform (Employee) .............................................................................................. 30
Figure 12: SatisFactory Platform (Platform Presentation, Summary, Challenges) ....................................... 30
Figure 13: SatisFactory Platform (Manager) ............................................................................................... 31
Figure 14: SatisFactory Platform (Game Processes) .................................................................................... 31
Figure 15: Architecture overview of the Social Interaction Platform. The front end on top communicates with the back end on the bottom ...................................................................................... 32
Figure 16: Social Collaboration Platform, Login Page .................................................................................. 36
Figure 17: Social Collaboration Platform, Registration Page ....................................................................... 36
Figure 18: List of various types of employees ............................................................................................. 37
Figure 19: Social Collaboration Platform, Personal Summary Page ............................................................. 38
Figure 20: Gamification Status Page ........................................................................................................... 38
Figure 21: Gamification Awards Page ......................................................................................................... 39
Figure 22: Gamification SatisFactory Components’ games.......................................................................... 39
Figure 23: Personal Details Page ................................................................................................................ 40
Figure 24: Types of Employees .................................................................................................................. 40
Figure 25: List of Nationalities ................................................................................................................... 41
Figure 26: Calendar View for DOB .............................................................................................................. 41
Figure 27: Action Wall (Videos, Photos of me) ........................................................................................... 42
Figure 28: View Photo Information ............................................................................................................. 42
Figure 29: Edit / Delete Video ..................................................................................................................... 43
Figure 30: Edit Video ................................................................................................................................... 43
Figure 31: Edit Photo ................................................................................................................................... 44
Figure 32: Delete Video ................................................................................................................................ 44
Figure 33: Delete Photo ................................................................................................................................ 45
Figure 34: Add a Video ......................................................... 45
Figure 35: Add a Photo .......................................................... 46
Figure 36: View all Connections .............................................. 46
Figure 37: View Connections Filters ........................................ 47
Figure 38: Add a Post ............................................................... 47
Figure 39: View Posts .............................................................. 48
Figure 40: Like, Add, Share a Post .......................................... 48
Figure 41: Publish a Post .......................................................... 49
Figure 42: Add a Video (Insert YouTube Link) ......................... 49
Figure 43: Add a Video (Upload To Youtube) ............................ 50
Figure 44: Add a Video (Local Upload) ..................................... 50
Figure 45: Edit, Like a Post ......................................................... 51
Figure 46: Share a Post .............................................................. 51
Figure 47: Searching Tool .......................................................... 52
Figure 48: Employee Type Filter ............................................... 52
Figure 49: Follow/Unfollow, Connect/Disconnect ....................... 53
Figure 50: View All Questions .................................................. 54
Figure 51: “Recently Added”, “Popular” Filters ......................... 54
Figure 52: Votes, Edit a Question, Post an Answer ...................... 55
Figure 53: Answer a Question ..................................................... 55
Figure 54: Post Your Question ................................................... 56
Figure 55: My Activity ............................................................... 56
Figure 56: Protect your Health Hardware Architecture .................. 58
Figure 57: Electrodes inside the soap dispenser ......................... 59
Figure 58: Soap dispenser connected to Raspberry Pi .................. 59
Figure 59: Protect your Health software flow chart ...................... 61
LIST OF TABLES

Table 1: Gamification API ................................................................. 18
Table 2: REST interfaces of suggestions server ........................................... 26
### LIST OF DEFINITIONS & ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ARM</td>
<td>Advanced RISC Machines</td>
</tr>
<tr>
<td>CIDEM</td>
<td>Common Information Data Exchange Model</td>
</tr>
<tr>
<td>CRUD</td>
<td>Create, Update, Delete</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>FW</td>
<td>Framework</td>
</tr>
<tr>
<td>GPIO</td>
<td>General-purpose Input/Output</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphics Processing Unit</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HDMI</td>
<td>High Definition Multimedia Interface</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>I2C</td>
<td>Inter-Integrated Circuit</td>
</tr>
<tr>
<td>ID</td>
<td>Identifier</td>
</tr>
<tr>
<td>JS</td>
<td>JavaScript</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MIPI</td>
<td>Mobile Industry Processor Interface</td>
</tr>
<tr>
<td>MIT</td>
<td>Massachusetts Institute of Technology</td>
</tr>
<tr>
<td>OpenELEC</td>
<td>Open Embedded Linux Entertainment Center</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>OSMC</td>
<td>Open Source Media Center</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>SD Card</td>
<td>Secure Digital Memory Card</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver Transmitter</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modelling Language</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>
EXECUTIVE SUMMARY

The present document is a deliverable of the SatisFactory project, funded by the European Commission’s Directorate-General for Research and Innovation (DG RTD), under its Horizon 2020 Research and innovation programme (H2020), reporting the results of the activities carried out by WP3. SatisFactory aims to develop an ecosystem of innovative technological components that would assist the daily operations of the people working at industrial environment. More specifically the developments of SatisFactory will be demonstrated to three diverse shop floors from discrete manufacturing (COMAU), batch products manufacturing (SUNLIGHT Systems) and continuous processes (CERTH/CPERI).

The deliverable D3.4 documents the complete implementation details of a few of the perceived concepts that were selected for elaboration. The details of the entire set of concepts and how they evolved are documented in D2.3 - Social Experience and Gamification techniques for increasing attractiveness. In addition, the deliverable D3.2 – Situated and Attractive Information Exchange Techniques for Workers, describes the design details of the selected few of these concepts. The implemented concepts include The Gamification Framework, the Suggestions for Improvement Platform, The Social Interaction Platform and the Protect your Health System. In addition to these concepts, this deliverable also describes the use of the Digital Andon System as the visualizer of the cumulative scores for the Gamification Framework.

D3.4 documents which technologies were used in order to implement the above mentioned systems. Also, what approach was followed for implementing these concepts. It explains the thorough details of the architecture of these systems. Additionally, it describes the details of the REST calls provided by the Gamification Framework API, explaining how various SatisFactory components can communicate with the Gamification Framework and become a part of the whole gamified environment.
1 INTRODUCTION

Task T3.4 deals with the implementation of collaborative applications aiming to support work processes in an attractive and comfortable way. The task is in close relation to tasks T2.3 and T3.2. In task T2.3, concepts for the enhancement of the working environments are worked out, based on the results of the domain analysis and formulated requirements in WP1. In total, 19 different ideas were worked out. A few of them were selected for elaboration. Finally, four of them were selected for implementation: The Gamification Framework, the Suggestions for Improvement Platform, the Social Interaction Platform and the Protect your Health System. This text documents the respective software deliverables. The conceptual work behind it is described in D2.3 - Social Experience and Gamification techniques for increasing attractiveness. T3.2 focuses on the user interface parts. This is described in D3.2 - Situated and Attractive Information Exchange Techniques for Workers.

The Suggestions for Improvement Platform and the Social Interaction Platform are both tools to support collaboration between workers inside the factory. Thus, they increase the professional, but also the personal connection between workers. The Suggestions Platform aims at improving the information flow between workers and decision makers; so that it is assured that worker suggestions reach the responsible decision maker who takes the decision whether to implement it, to implement it with modifications or to reject it. Finally, and most importantly, it guarantees that this decision is fed back to the worker. The Social Interaction Platform can be used to exchange knowledge but also social content through sound, video or text.

This fun related aspect is supported by the Gamification Framework. The gamification parts shall motivate workers for all actions that are not performed often or well enough, presumably because they are unpopular for some reason. Gamifying these tasks shall raise them popularity. At the same time, the overall gamification approach shall add a fun aspect to work and help increasing the overall attractiveness of the workplace.

Any task on the shop floor can be gamified. For this, existing or new systems need to detect these tasks and connect them to the Gamification Framework. One example is the Protect your Health System, which aims at encouraging workers to wash their hands more properly after having worked with lead. The gamification approach wants to combine improved worker hygiene, thus health, with increased fun and satisfaction. The Protect your Health System is also a good example of the ubiquitous computing concept applied for the developed systems. One core concept of ubiquitous computing is calm computing. This means to minimize the perceptible invasiveness of computers in users’ well-known environment. Therefore, instead of adding computers, which have to be explicitly handled, the Protect your Health System introduces computerized soap dispensers (described in section 5.2.1.1), which can be used in the same way as their traditional counterparts. The user implicitly interacts with a computerized system, without noticing it. This is also referred to as integration of the virtual into the real world.
2 IMPLEMENTATION OF GAMIFICATION FRAMEWORK

2.1 APPROACH AND USED TECHNOLOGIES

The Gamification Framework is meant to be a central framework. All the external Satisfactory components communicate with this framework via REST calls. The Gamification Framework is implemented using the technologies briefly introduced in the next subsections.

2.1.1 RESTful web API

The representational state transfer (REST) is the software architectural style of the World Wide Web. It consists of a coordinated set of components, connectors, and data elements within a distributed hypermedia system, where the focus is on component roles and a specific set of interactions between data elements rather than implementation details. Its purpose is to induce performance, scalability, simplicity, modifiability, visibility, portability, and reliability.

2.1.2 Node.js

Node.js is a server-based platform for distributed applications. It is often used for implementing web servers. Applications are developed in JavaScript. Node.js is MIT-licensed, which allows free usage. It is very lightweight in terms of storage because of its event-driven architecture. This gives a very fast performance. Node.js is available for Mac, Windows and UNIX. It is a widely used technology.

The Node.js binary already includes some basic modules. Further modules can be installed, updated and resolved over its package manager npm (Node Package Manager). There is very active community support so that thousands of modules are available.

2.1.3 CIDEM

The Common Information Data Exchange Model (CIDEM) aims to provide a model of information elements (e.g. concepts, even, relations, interfaces) used for information exchange between components as well as for modeling work performed by other tasks. The CIDEM definition is considered as a shared vocabulary that enables to address the information needs for the SatisFactory framework components.

2.2 IMPLEMENTATION

2.2.1 Data model

Figure 1 shows the data model for the Gamification Framework. As the Gamification Framework communicates with the SatisFactory repository via the middleware, the data model has to be CIDEM compliant.
Figure 1: CIDEM Compliant Data Model Schema for the Gamification Framework
2.2.2 Architecture

Figure 2 shows the architecture of the Gamification Framework. The figure clearly shows how the idea is to have a centralized architecture, with at the centre Gamification Framework. Different SatisFactory components can use the APIs to become a part of the whole gamification environment at the factory. All the components use REST calls to communicate with the Gamification Framework. The next section contains the details of the API provided to various SatisFactory components to communicate with the Gamification Framework, which communicates with the SatisFactory repository via the middleware.

Figure 2: Gamification Framework Architecture

2.2.3 API

Figure 3 shows how an external component e.g. Suggestions for Improvement platform communicates with the Gamification Framework. One of the gamified activities in Suggestions for Improvement platform is to submit a suggestion. Therefore, when a worker submits a suggestion, the Gamification Framework is informed via a push call and the corresponding points are updated. In order for the worker to view these points, he can access the personal point visualizer i.e. social collaboration platform. The personal point visualizer makes a get call to obtain the updated points for that particular worker.
Table 1 contains the details of the API provided to all the satisfactory components in order to enable them to communicate with the Gamification Framework.

**Table 1: Gamification API**

<table>
<thead>
<tr>
<th>URL</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gaming/games</td>
<td>GET</td>
<td>Returns a list of all games currently registered in the Gamification Framework, consisting of game Id, description and registered tasks. Tasks consist of task Id and score, which tells how many points the task is worth.</td>
</tr>
<tr>
<td>gaming/games</td>
<td>POST</td>
<td>Adds a new game to the list of games. Request Body: JSON Payload - New game</td>
</tr>
</tbody>
</table>

```json
{
    "games": [
        {
            "gameId": "suggestions",
            "desc": "suggestions game",
        }
    ]
}
```
"tasks": [
  {
    "taskId": "create",
    "score": 4
  },
  {
    "taskId": "vote",
    "score": 1
  }
]

gaming/games POST Registers a game under ID (gameld) with description (desc) at the Gamification Framework

  Request Body: gameld, desc

gaming/games/:gameld DELETE Deletes a game (gameld) from the Gamification Framework

gaming/games/:gameID/tasks POST Adds a task with a task id (taskId) to a game (gameld) and tell how many points (score) the task shall be worth

  Request Body: taskId, score

gaming/games/:gameld/tasks/:taskId PUT Modifies a task with taskId associated to a game (gameld) and tells how many points (score) the task shall be worth. The old score is overwritten with score

  Request Body: score

gaming/games/:gameld/tasks/:taskId DELETE Deletes a task with taskId from a game (gameld)

gaming/players GET Returns a list of all players registered to the framework. The list consists of playerId and scores. Scores shows the score of the particular player for each game

gaming/players/:playerId/games/:gameld/tasks/:taskId/push PUT Informs the Gamification Framework that a particular player (playerId) has performed a particular task (taskId) in a particular game (gameld). This makes the Gamification Framework recalculate the points, etc. The components must make this call every time they detect that a gamified action is performed

gaming/players/:playerId/scores GET Returns the score of the particular player (playerId) for each game
2.2.4 Visualizing Team Point with Digital Andon System

Figure 4 shows how the Gamification Framework uses the digital Andon system to visualize the team points. The digital Andon system is described in Chapter 6 of Deliverable D2.3 - Social Experience and Gamification techniques for increasing attractiveness. The visual design of the team points is described in Section 4.2 of Deliverable D3.2 - Situated and attractive information exchange techniques for workers.

After starting the Gamification Framework, it calls the digital Andon system for some initial actions. First, a new screen has to be created. The digital Andon returns a screen ID for this call. The Gamification Framework remembers this ID and refers to it in all following calls. As next step, the
background is created, i.e. all static visual elements. Then for each of the three scores, a text field is created and again the returned text field IDs are saved for referring to them each time the scores are changed. Finally, this initial screen is released.

After the initial steps, the system is continuously running until it is manually stopped. There are two events, which can trigger further actions. The first event is that a connected system notifies the Gamification Framework about a gamified task being performed. Then, the Gamification Framework adds the points associated to that actions to the team point counter and updates the today’s team point score. If the today’s team points are higher than yesterday’s score, it means that today’s goal has been reached and the win icon needs to be displayed. Afterwards the updated view is released.

The second event that triggers further actions is the beginning of a new day. First, it is checked whether the score of the ending day exceeded the highest score and if yes, it is set as new high score. After that, it must be checked whether the goal has been reached, i.e. if the today’s score is higher than the yesterday’s score or not. If yes, today’s score becomes the new yesterday score, being the goal to reach for the next day. If no, a new win streak is starting, so yesterday score is reset to zero. Finally, the today score is reset to zero and the updated view is released.
3 IMPLEMENTATION OF SUGGESTIONS PLATFORM

3.1 APPROACH AND USED TECHNOLOGIES

First, an introduction into the technologies, which are used for implementing the suggestions for improvement platform is given.

3.1.1 Node.js
Node.js is also used by the Gamification Framework and as such described in Section 2.1.2. The server part of the suggestions for improvement platform is implemented in Node.js.

3.1.2 Ionic
Ionic is a framework for developing hybrid mobile applications based on HTML5, CSS and JavaScript. It allows developing non-native apps that retain a native look and feel and also have full access to all functionalities available for native apps. Ionic is built on top of AngularJS and Cordova/Phone Gap, which suits our purposes of reusing most of our code base for the MyFactoryManagement app, which will be an AngularJS web application.

The mobile kiosk application will be implemented using Ionic.

3.1.3 AngularJS
AngularJS is a JavaScript-based web application framework. AngularJS key feature is the two-way databinding used to bind application data to an HTML template. Contrary to some other JavaScript frameworks, AngularJS keeps the application logic decoupled from the DOM manipulation; this improves the testability of the code as well as the overall performance.

As mentioned above, the MyFactoryManagement web app will be implemented using AngularJS, taking advantage of the connection between AngularJS and Ionic.

3.2 IMPLEMENTATION

3.2.1 Architecture

Figure 5 shows the architecture of the suggestions for improvement platform in a UML component diagram.
The application MyFactory serves as user interface for workers. Most triggered actions are forwarded to the server backend, such as creating, updating or viewing suggestions, up- or down voting them or assigning categories to a suggestion.

MyFactoryManagement has the same functionalities as MyFactory. In addition, suggestions statuses can be changed, i.e. suggestions can be accepted, accepted with modification or rejected. For this reason, a login mechanism to authenticate the user is necessary, to trace the operations and handle different levels of privileges. In addition, other decision makers can be administrated. Finally, another additional feature is the possibility to add, view or accept collaborations for a suggestion.

Accounts are authenticated and authorized by the *User Management* component. The CIDEM repository is used as database for suggestions, categories, votes and collaborations.

Data model

Figure 6 shows how the data model of the suggestions platform was initially implemented.
After the third iteration of the suggestions GUI (cf. D3.2 - Situated and Attractive Information Exchange Techniques for Workers, Section 3.2.5), several changes cause to update the data model as well. The main changes are:

- Photos (and other media) were removed since it will not be possible from the kiosk
- Comments were removed - We will just use up-/down voting
- Collaborations were simplified

The result is depicted in Figure 7.

Figure 6: First version of the suggestion platform data model

Figure 7: Second version of the suggestion platform data model
3.2.2 Activity Diagram

Figure 8 and Figure 9 show how the system works. The normal flow is that the user triggers a certain action, such as submitting a suggestion, view suggestions details or up- and down voting. The GUI receives the request, triggers in turn actions at the server or collects data and notifies back to the user.

Figure 8: Activity diagram suggestions, first part

Figure 9: Activity diagram suggestions, second part

3.2.3 Server API

The server provides the following RESTful interfaces for the client to call (cf. Table 2). The table also shows for which user groups this is needed. For the identified user groups, refer to Section 5.1.3 of D2.3 - Social Experience and Gamification techniques for increasing attractiveness.
3.2.4 **Connection to Gamification Framework**

The suggestions platform is connected to the Gamification Framework, i.e. workers can collect points for certain events which occur in the suggestions platform.

Three events from the suggestions platform are gamified:

- Submitting a suggestion
- Submitted suggestion gets accepted
- Up voting a suggestion

Initially, a game with these tasks was created at the Gamification Framework using the REST calls specified in Section 2.2.3. Every time such an event occurs, the Suggestions platform calls „push“.

3.3 **Deployment**

Figure 10 shows how the suggestions platform is deployed.
The MyFactory app is running on a tablet computer. This tablet is installed on the shop floor as a kiosk, meaning that it is fixed to a wall or on a floor-stand, in an area where every worker can access it. Data exchange with the backend is working over Wi-Fi. The backend is installed on a server in the factory. It uses the CIDEM repository for storage and accesses the User Management component, which runs on the same server.

MyFactoryManagement is a web application, which can be accessed with any browser. It is meant to be used by decision makers, most probably from their offices. Nevertheless, it is also possible to access it from a mobile device, e.g., from a smartphone while the decision maker is at the shop floor. The device that is used for accessing it just has to be inside the factory network so that it can access the MyFactoryManagement web application on the server. This deployment configuration ensures that it can only be accessed from within the factory, which is a data security feature.
4 **IMPLEMENTATION OF SOCIAL INTERACTION PLATFORM**

4.1 **APPROACH AND USED TECHNOLOGIES**

4.1.1 **Technologies**

The Social Interaction Platform is a web application that follows the server-client architectural paradigm and therefore, utilizes web development frameworks that enable this approach. The server-side of the platform is built with the Liferay v6.2 content management framework, which allows data storage, application logic implementation and web service deployment. The client-side of the platform, which provides a Graphical User Interface to the end-user, is developed with HTML5, CSS3, the Angular v1.5.6 web application framework and the Angular Material v1.0.9 User Interface component framework. The Angular framework allows the rapid development of the front-end as a single-page application where some of the application logic is implemented client-side and data manipulation is achieved by consuming web services exposed by the platform’s back-end. This way the end-user never has to reload the web page and thus gets a more fluid user experience as if he was using a desktop application.

4.1.1.1 **Liferay**

The Liferay content management framework offers developers the ability to rapidly develop web applications using popular technologies such as the Java programming language, the MySQL database server and RESTful web services via the Tomcat application server. Java is a high-level, object-oriented, cross-platform programming language that features a large standard library and the ability to build web applications that can be deployed by Java application servers. Developers use Java to build portlets, which can be thought of as Java web applications, containing application logic and access to database data that is exposed via RESTful web services when the portlet is deployed on Tomcat. While Liferay may also be used to develop the front-end of a web application with the use of Java server pages, in the case of the Social Interaction Platform the portlet only contained web services that enabled access to the back-end. Liferay provides a mechanism where the developer describes the application entities in XML format and subsequently the corresponding tables are auto-generated in the database and Java classes that provide basic CRUD (Create, Read, Update, Delete) operations on these tables are also auto-generated. By utilizing the aforementioned classes, the developer is able to write the application logic in the web services without having to deal with the database directly. This mechanism also allows Liferay to employ a cache of database data that significantly increases the access speed to the data, especially in heavy usage scenarios. Liferay also features premade entities, such as users, user groups, categories, social relationships, calendar events, entities for messaging and forum implementation, etc., which are common in web applications and thus expedites the development effort.

4.1.1.2 **Angular**

The Angular framework is designed to facilitate the development of single-page web applications and overcome the shortcomings of HTML when displaying dynamic content. It enables developers to
build web applications where the client-side follows the model-view-controller design pattern where presentation, data and logic components are separated. This is made possible by Angular being able to provide two-way data binding between model and view, which means that the view is updated whenever the model changes and the model is updated whenever the view changes. Two-way data binding allows view templates to exist in client-side code only as data are fetched from the back-end via web services, thus relieving the server-side of this responsibility. The controller contains JavaScript code where the behavior of the view (HTML page) is defined by writing in a clean readable form without the usual approach of manipulating the HTML DOM (Document Object Model) and listening for changes. The view contains more than regular HTML code because Angular allows the custom extension of the HTML language by providing a feature called Directives. Directives let you define custom HTML syntax by placing markers on a DOM element, which attach a specific behavior to that element or may transform that element and its children. Angular comes with pre-build directives to address basic functions, but most importantly it allows the developer to create his own directives that represent reusable components that hide complex DOM structure and behavior. This feature renders Angular an open and extensible framework with multiple third-party plugins that enrich the functionality of a web application built with the Angular framework.

4.1.1.3 Angular Material
Angular Material is a User Interface Component framework for web applications built with the Angular framework. It is also a reference implementation of Google's Material Design specification, providing a set of reusable, well tested, and accessible UI components based on Material Design. Material Design is a specification for a unified system of visual, motion, and interaction design that adapts across different devices and different screen sizes. Material Design makes use of grid-based layouts, responsive animations and transitions, padding, and depth effects such as lighting and shadows.

4.1.2 Content
The satisfactory platform includes the following main features according to the users and social gamification requirements (D2.3):

1. Social network features
2. Gamification elements
3. Tips & tricks tool

The current content of the SatisFactory platform is presented on the following diagrams. The empowered employee can manage the game features that workers can participate. (Figures Figure 11, Figure 12, Figure 13, Figure 14)
Figure 11: SatisFactory Platform (Employee)

Figure 12: SatisFactory Platform (Platform Presentation, Summary, Challenges)
4.2 IMPLEMENTATION

4.2.1 Architecture

The Social Interaction Platform’s architecture follows the client-server model, where the front-end part (client-side) contains the User Interface and the back-end part (server-side) is responsible for the business logic, data storage and data manipulation. The front end consumes web services, exposed by the back end, which allow it to retrieve the data that are presented to the user and to store the data that the user wants to upload to the platform. The front end consists of the following components: Network, Newsfeed, Tips & Tricks, Profile, Incidents, Notifications, Multimedia, Messenger, Settings. The back end consists of three layers. The top layer is the Remote API layer where all the web services are defined by declaring name, arguments, return type, authorization policy and the specific function call that implements the business logic. Below the Remote API layer, is the Business layer that contains all the code written by the developers consisting of the functions that implement the web services as well as any other helper functions and utility classes. Below the Business layer is the Entities layer. The Entities layer contains an XML description of all the
developer-defined entities and the auto-generated code that the Liferay platform produces based on the aforementioned XML description. The developer may define in XML an entity such as (e.g.) the “User” entity that describes a user of the Social Interaction Platform. Afterwards, the Liferay platform can be used to generate a table for the “User” entity in the system’s database along with Java classes that contain pre-made functions that handle CRUD operations for the “User”. This way the developer does not have to deal directly with the system’s database, but he can call the aforementioned functions from the Business layer in order to manipulate the various entities in the code. The Entities layer handles most of the communication with the system’s database and the filesystem where multimedia files are stored. It should also be noted that the business layer connects to a LinkSmart system via a publish/subscribe mechanism in order to receive events that describe incidents occurring on the shop floor. This information is processed and forwarded to the client so it can be displayed to the end user.

![Architecture overview of the Social Interaction Platform](image)

**Figure 15:** Architecture overview of the Social Interaction Platform. The front end on top communicates with the back end on the bottom.

### 4.2.2 Back End

#### 4.2.2.1 Users

The user is a core entity in the Social Interaction platform, as it is expected in a social application. The Liferay platform provides a customizable premade User entity, which is used as the basis of the Social Platform user. The premade User entity contains useful fields such as name, email, gender, birthday, etc., but in order to add extra information to the entity, which is necessary for the Social Platform, such as type, nationality, address, etc., it was necessary to add the corresponding custom fields to the entity. Custom fields is a mechanism provided by Liferay that allows the developer to add extra fields of information to pre-existing Liferay entities, such as users, events, messages, etc. A number of web services were implemented to provide for basic user functionality such as login, logout, register, update user, get user, upload profile image, search users, etc.
4.2.2.2 **Social Connections**

Two types of social connections are available in the Social Interaction Platform: co-worker and follower. The co-worker is roughly the equivalent of the more common “friend” relationship in traditional social networks. It should be noted that these user relationships are natively supported by the Liferay platform through premade functions that establish/break relationships between User entities, fetch related Users for a specific user, fetch Activities of related Users, etc. Establishing the co-worker relationship with another user allows one more access to that user, such as viewing posts and personal information that is meant for co-workers only or being able to send messages to that user, etc. In order to establish the co-worker relationship with another user, a request must be sent and then are accepted by that user. The follower relationship allows mainly one user to follow the activities of another user. This means that whenever a user posts something publicly available on his timeline, that post also appears in the newsfeed of all the users that follow him. Contrary to the co-worker connection, the follower connection is established immediately without requiring acceptance from the target user. The following functionality is supported by the implementation of the corresponding web services: request co-worker connection, accept co-worker connection, cancel co-worker connection, get pending co-worker requests, add follower, remove follower, get connections, search connections, etc.

4.2.2.3 **Multimedia**

The Social Interaction Platform supports the use of multimedia such as text, images, local videos and YouTube videos. The user may post multimedia content on his own timeline or on other users’ timeline and he may also post an image or a video on the Q&A forum. The functionality for doing text posts and local multimedia posts is aided by the premade BlogEntry and Document entities provided by the Liferay platform, the latter’s intended use is for uploading and storing multimedia files to the Liferay server. Every post, regardless of type, has title, description and an access attribute indicating whether it is public, private or for co-workers only. The images and videos may also be tagged with various user names, indicating that there is a connection between the tagged user and the medium. The tagged user has to decide whether to accept or reject the tagging. Regarding the videos, the user can choose to upload a video locally to the Social Platform back-end, upload the video to a YouTube account or insert a link of an existing YouTube video. Posted content can be edited, namely its title, description, access qualifier, and tagged users, only by the user who uploaded it. The user may also view all the publicly accessible videos and images uploaded to the platform as well as search among them by title or description. Web services implemented to support multimedia functionality include the following: post text, post local multimedia, post YouTube video, upload and post YouTube video, delete local multimedia, delete YouTube video, accept tagging, reject tagging, edit text post, edit local multimedia, edit YouTube video, get multimedia, get multimedia of tagged user, search multimedia, search tagged multimedia, etc.

4.2.2.4 **Activities**

Everything that shows up on a user’s timeline or newsfeed is a user activity. There are activities for posting content on a timeline, for changing profile photo, for posting questions and answers on the Q&A forum and for sharing an existing activity. The Liferay platform provides the premade Activity entity, intended exactly for this usage, as it is evident by the fact that when adding a text post or uploading a multimedia file, Liferay automatically creates the corresponding Activity. Liferay also provides library functions that fetch the Activities of a certain user or a group of related users, etc.
The developer may create an instance of an Activity to be associated with custom class instances, so this mechanism is utilized to associate Activities with the Social Platform’s custom activities such as questions and answers of the Q&A forum. The creator of an activity, e.g. the user who posted content on his timeline, has the ability to change the title, description and date of the activity and delete the activity. A user may like or dislike an activity, post/delete a comment on an activity, like or dislike an activity comment, and share an activity on his/a friend’s timeline.

4.2.2.5 Messaging

The Social Platform supports the exchange of messages between users, a functionality whose development was aided by an existing Liferay platform component that offers similar functionality. Messages belong to conversation threads, created when a user sends a message to a user or a group of users that he has never sent one before. Sending a message to user/group of users who already participate in another discussion thread, results in adding the new message to the existing thread. Messages arrive instantly to the recipients who are online, at the time that the message was sent, thus enabling a sort of live chat functionality. This is achieved by employing a library that implements the WebSocket protocol. The WebSocket protocol provides a full-duplex communication channel over a single TCP connection, thus allowing the web server (Liferay/Tomcat) to send data to the web browser over the default TCP port. Web services of the Social Interaction Platform that allow the messaging functionality include: send message, get user messages, set thread read, etc.

4.2.2.6 Notifications

Notifications of three types are supported by the Social Interaction Platform: social notifications, achievement notifications and incidents. Social notifications are sent to a user when other users comment on his posts, like his posts, comment on a post he commented, tag him on a video/photo, share content with him, start following him, accept his co-worker request or like his comments. Achievement notifications include notifications for events such as advancing a level, winning a badge, climbing the rank, etc. Incidents are notifications that are generated whenever an incident occurs at the shop floor, and the Social Platform is notified via a connection with a LinkSmart server instance, employing a publish/subscribe mechanism. All of the above notifications are stored in the system’s database and then propagated to online users via the WebSocket protocol, as it was discussed in the previous section. Some of the web services that offer notification functionality are: get latest notifications, set notification read, get incidents, view incidents, etc.

4.2.2.7 Settings

The Social Interaction Platform features various settings that customize its functionality and have different values per user in order to address the needs of each user. There are three main categories of settings: account, privacy and notification. Account settings include the user’s first and last name, the user’s email and password, and the user’s language and time zone. The above settings are pre-existing or custom fields of the User entity, so there was no need to implement something extra to cover this functionality. The privacy settings regard the privacy of a user’s personal information, of his timeline, of his posts, and his general presence on the Social Platform by allowing the restriction of users that can send him a message, tag him on a video, request a co-worker connection, etc. The notification settings allow a user to enable/disable the notifications he receives either in-site or via email, for the various cases where the Social Platform sends notifications, as discussed in the previous section. The privacy and notification settings had to be implemented from scratch, so the following mechanism was built. A “Setting” entity was created in order to model a setting of the
Social Platform, with fields for id, name, type, values, group and default value, while an entity “UserSetting” was created to hold the user’s choice for a particular setting, with fields for setting-id, user-id, and current value. Whenever a user changes a setting, the corresponding UserSetting instance is created or updated. If there is no such instance of UserSetting when it is required for application functionality to take settings into consideration, the default value from the corresponding Setting entity is used. This mechanism allows settings to be modified easily and makes the addition of new settings a trivial matter.

4.2.2.8 Q&A Forum

The Social Platform back end features a Q&A forum, which is presented to the user in the Tips & Tricks component of the front end. There are web services that enable a user to post questions and answers to those questions with multimedia content. The user is also able to up-vote or down-vote questions and answers alike. The various questions are categorised in different topics, topics that can be modified by the forum’s moderators. Each question has a title, a body, an author, a creation date a number of positive votes, a number of negative votes and a number of unique views per user. Answers have a body, an author, a creation date, positive votes, and negative votes, while users may also post comments on the answers. Users may edit or delete their own questions and answers, and they may also cancel their vote on a question or answer.

4.2.3 Front End

In this chapter, current functionalities of the platform are presented.

4.2.3.1 Welcome page

After registering in the platform user can login in order for the content to be personalized. (Figure 16). When clicking “register” button the user fills name, email, password, gender, type and date of birth. (Figure 17) A pre-defined list of types of employees have been designed according to the factory needs. (Figure 18)
Figure 16: Social Collaboration Platform, Login Page

Figure 17: Social Collaboration Platform, Registration Page
4.2.3.2 **Personal Page**

Tips & tricks provide the possibility to the users to ask and offer help. Network offers the possibility to the users to grow their network by viewing and adding co-workers. Home button gives the possibility to the users to view the network activities. This means that the users can view users’ activities once they are posted (timeline). My profile button leads the user to the personal summary page. Once the user is logged in, he is notified for incidents, gamification news, co-workers requests, messages and general news of the network. (Figure 19)
4.2.3.3 **My Profile page**

Achievements consists of four sub sections: status, awards, challenges, in the crowd. Status gives the information that the user needs concerning the points, level, games and tasks completed (Figure 20). Awards section inform analytically the user in regards with the awards earned (Figure 21). Challenges is the games available for the user to participate. In this section user can join a game or leave clicking on the “join button” or “leave button” respectively (Figure 22).
In “Personal” section (Figure 23) user can fill and edit the personal information such as cv, type of employee (Figure 24), gender, nationality (Figure 25), date of birth (Figure 26), residence, e-mail and phone number.
**Figure 23: Personal Details Page**

**Figure 24: Types of Employees**
Action Wall consists of videos and photos of the user (Figure 27). Whether the user views videos or photos with a mouse over can view information like title and views (Figure 28), as well as can edit or delete current video or photo (Figure 29-Figure 33). By clicking on “Add” button user can upload a video if the current section is “video of me” and a photo if the current section is “photos of me” (Figure 34 and Figure 35). User can upload a video from YouTube, from pc and upload a video on YouTube.
Figure 27: Action Wall (Videos, Photos of me)

Figure 28: View Photo Information
Figure 29: Edit / Delete Video

Figure 30: Edit Video
D.4.4 - Collaborative platform for work process support | August 2016 | FIT
SatisFactory project | GA #636302

Figure 31: Edit Photo

Figure 32: Delete Video
Figure 33: Delete Photo

Figure 34: Add a Video
Achievements, Personal, Action Wall, Connections, Timeline

In connections section user can view personal connections, followed and connected co-workers (Figure 36). Also can find connections through search tool and filter the results viewing all connections, co-workers, following and followers (Figure 37).

Figure 36: View all Connections
Figure 37: View Connections Filters

Personal timeline shows the users actions. User can add a post, which can be text, video or photo (Figure 38).

Figure 38: Add a Post

4.2.3.1 Home page

In this section, a timeline is available which consists of the posts that the user’s network publishes. User can view posts (Figure 39). Clicking on them user can view details like “like”, “share” and “add a comment” (Figure 40). Like on personal timeline post with text video and photo is available. Posts
can be published on co-workers, public, or private (Figure 41). Video can be uploaded from YouTube or pc and can be uploaded on YouTube or in the platform (Figure 42- Figure 44). A post consists of its title, description, date & time. User can edit the post, add a comment, delete, edit or like a comment. Also, one can share the whole post or like it (Figure 45). Clicking on “share” button, the user can publish the post to the personal timeline, to a co-worker’s timeline or can send it via messenger (Figure 46).

Figure 39: View Posts

Figure 40: Like, Add, Share a Post
### Figure 41: Publish a Post

#### Post text

**Title:**

Need help - shop floor

**Body:**

Need a hand in shop floor to lift a package. By error handling the calibration must be done manually; 2 pairs of hand needed.

**Privacy:**

Public

### Figure 42: Add a Video (Insert YouTube Link)

#### Add a Video

**Video URL:**

https://www.youtube.com/watch?v=YHc05ULJLtg

**Title:**

Best Hans Zimmer Music (Top 10 HD)

**Description:**

- Subscribe: https://www.youtube.com/channel/UCyqQ2_d_fXIJOo389N_BHqmg
- More Videos:
  
  https://www.youtube.com/channel/UCyqQ2_d_fXIJOo389N_BHqmg

**Users tagging:**

- Enter a name

**Privacy:**

Public

---

D3.4 - Collaborative platform for work process support ▪ August 2016 ▪ FIT

SatisFactory project ▪ GA #636302
Figure 43: Add a Video (Upload To Youtube)

Figure 44: Add a Video (Local Upload)
4.2.3.1  **Network page**

In this section, the user can grow his personal network by searching for co-workers to connect or follow. For this reason, a searching tool is available (Figure 47). A filter referring to the type of
employee facilitates the user’s searching (Figure 48). By clicking on the right button, the user can follow co-workers, connect with them, unfollow and disconnect (Figure 49).

Figure 47: Searching Tool

Figure 48: Employee Type Filter
4.2.3.2 **Tips & Tricks page**

In this section, the user can view all of the questions, ask a question and view personal activity concerning tips & tricks (Figure 50). In the Questions section can view a question using the searching tool and the filter viewing the “recently asked” questions or the most “popular” (Figure 51). By clicking on a question, the user can view the title and the description of the question, the votes, and the answers and can edit the question or answers, vote or comment an answer (Figure 52). Additionally, the user can also add an answer on the current question (Figure 53). In “Ask a question” section, the user can post a question in tips & tricks writing a title, a description and choosing a topic for the question (Figure 54). In “My activity” section, the user can view personal questions and answers with their features (Figure 55).

![Figure 49: Follow/Unfollow, Connect/Disconnect](image)
Figure 50: View All Questions

Figure 51: “Recently Added”, “Popular” Filters
Figure 52: Votes, Edit a Question, Post an Answer

Figure 53: Answer a Question
Figure 54: Post Your Question

Figure 55: My Activity
5 PROTECT YOUR HEALTH

5.1 APPROACH AND USED TECHNOLOGIES

In this section, a short overview of the used technologies for the Protect your Health system will be given. The technologies are introduced in the following subsections.

5.1.1 Raspberry Pi

The Raspberry Pi is a single-board computer developed in the United Kingdom originally for teaching purposes. It gained a lot of popularity among software developers and engineers as an easy to use small computer for rapid prototyping projects. Today the Raspberry Pi boasts a large online community and eight different models are available. They all contain an ARM based processor, a dedicated GPU and at least 256 MB RAM. The Raspberry Pi provides several hardware interfaces including USB, HDMI, MIPI camera interface, a 3.5mm audio jack, several GPIOs, UART, I2C, SPI and a SD card socket. The SD card socket is mainly as flash storage for the execution of an operating system. The Raspberry Pi uses primarily Linux-kernel-based operating systems like Arch Linux ARM, OpenELEC, OSMC, Pidora, Puppy Linux, RISC OS and many more. The main operating system is Raspbian, a Debian-based Linux distribution. Raspbian is maintained and recommended by the Raspberry Pi Foundation. Like a standard computer the Raspberry Pi can be connected to a monitor, keyboard and mouse. For accessing the internet, a LAN interface is also available. Alternatively, a WLAN USB-stick can be used to access wireless networks.

5.1.2 Raspbian

Raspbian is a Debian-based computer operating system free and optimized for the Raspberry Pi hardware. It was initially built in June 2012 and developed for the Raspberry Pi. It is not associated with the Raspberry Pi Foundation but the foundation provides a Raspbian image as an officially supported operating system. It is a free OS optimized for the Raspberry Pi hardware.

5.1.3 Python

Python is a widely used high-level programming language conceived in the late 1980s. The remarkable online community provides several libraries for integration in custom projects. It is a freely usable and distributable, general-purpose, dynamic programming language developed under an OSI-approved open source license. It is designed to emphasize code-readability and supports multiple programming paradigms including object-oriented, imperative and functional programming. The design of python is influenced by a collection of 20 software principles that are listed in the ‘Zen of Python’. These principles include principles like:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
5.1.4  **Putty**
PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

5.2  **IMPLEMENTATION**

5.2.1  **Hardware**
This section explains how the hardware in the Protect your Health system is implemented. Figure 56 provides an overview of the Protect your Health hardware architecture. On the left hand side, the modified soap dispenser with its internal electronic is displayed. The soap dispenser is connected to the Raspberry Pi, which in turn is connected to the Gamification Framework. The user can access the framework with a standard Laptop. Details are explained below.

![Figure 56: Protect your Health Hardware Architecture](image)

5.2.1.1  **Modified Soap Dispenser**
The modified soap dispenser is a common soap dispenser for industrial use enhanced with customized electronic mounted inside the housing. The electronic is intended to detect when the button used to be served with soap is pressed. To achieve that two electrodes are positioned in order to touch each other allowing current flow from one electrode to the other. The electrodes are shown in Figure 57. As soon as the button in the front is pressed one electrode moves breaking the connection between these electrodes. Two wires which are connected to the electrodes are leading out of the housing to a Raspberry Pi which is able to detect when the electrodes inside the soap dispenser are disconnected (see Figure 58).
5.2.1.2 Raspberry Pi

The task of the Raspberry Pi in this setup is the detection of a button press and the publishing of this event over its integrated LAN interface. The two electrodes inside the soap dispenser are connected with cables to two pins of the Raspberry Pi (see Figure 57). One electrode is connected to pin 18 which is configured as an INPUT with an activated internal Pull Up resistor pulling it to HIGH state (3.3 V). The other electrode is connected to a ground pin. If the status of pin 18 is now queried by the Raspberry Pi, it will be LOW (0V) because the ground pin pulls the voltage level of this pin to 0 V. As soon as the button is pressed, the electrodes inside the soap dispenser get disconnected and the pull up resistor pulls pin 18 to 3.3 V. Querying the status of this pin would now return HIGH state. Pin 18 is frequently monitored and as soon as the button is pressed the Raspberry Pi detects it and sends a message via its LAN interface to the gamification API.

5.2.1.3 Interfaces

There are two hardware interfaces involved in the Protect your Health system. The first interface consists of two cables connected to the GPIOs of the Raspberry Pi for detecting a button press of the soap dispenser. The second interface is the LAN interface of the Raspberry Pi. This interface submits data with the help of web services to the gamification API.

The Raspberry Pi uses HTTP methods to perform PUT requests to the gamification API whenever the soap dispenser button is pressed. The API returns a status code and if it is ‘successful’ status, the gamification API uses this information for further processing.
5.2.2 Software

This section gives an explanation of how the software in the Protect your Health system is implemented.

The custom software for the Protect your Health system is written in python. The script is running on a Raspbian distribution and can be started remotely from any computer connected to the internet using a SSH connection to the Raspberry Pi. As an alternative, the script starts itself as soon as the Raspberry Pi is powered on and system has booted.

As seen in Figure 59, the python script is structured as follows: First, an initialization process is started. In this process, necessary libraries are imported, the GPIOs are configured, the corresponding Pull Up Resistors are activated, an internal counter is initialized, the ‘mustUpdate’ flag is set to TRUE and the ‘publishUpdate’ function is defined. The ‘publishUpdate’ - function executes the ‘PUT’ request to the gamification API which is handled by the ‘requests’ library.
After the initialization process, a never ending loop is executed. This loop runs as long as the script is running. In each cycle of this loop, the state of pin 18 (pin where the button is connected to) is frequently queried in an if-condition. If this pin goes into HIGH state (button is pressed) the ‘mustUpdate’-flag is set to TRUE. After that the status of this flag is checked. If it is set to TRUE, the ‘publishUpdate’ - function is called and the flag is reset to FALSE. Furthermore, the script is paused for two seconds for denouncing purposes to prevent the Protect your Health score from being increased e.g. when someone presses the button multiple times in a short duration.

5.2.3 **Connection to Gamification Framework**

Protect your Health is connected to the Gamification Framework, i.e. workers can collect points.
The tool only consists of one gamified action: Wash hands (represented by pulling soap from the
dispenser.). As the Protect your Health system does not have an authentication mechanism, the
points are only collected for the overall team points.

Initially, a game with this task was created at the Gamification Framework using the REST calls
specified in Section 2.2.3. Every time a wash hands event occurs, the Protect your Health system calls
“push”.
6 CONCLUSIONS

The main objective of the present deliverable is to document the implementation details of the collaborative applications aiming to support work processes in an attractive and comfortable way. This deliverable described what technologies were used during the implementation process and what approach was followed. The selected concepts include the Gamification Framework, the Suggestions for Improvement Platform, The Social Interaction Platform and the Protect your Health System.

D3.4 explained the implementation details and architectural design of various platforms designed to encourage the social interaction between workers in factories. For example, the social interaction platform aims to improve the social aspect of the working experience for the workers. The gamification framework aims to introduce a sense of playfulness and competition to stimulate collaboration and motivate workers for doing tasks better or more frequent. This deliverable also described how the Digital Andon System can be used to visualize the cumulative score from the Gamification Framework.

Conclusively, this deliverable documented the implementation details of various systems and the technologies used in the implementation process. The concepts behind these systems and some others are documented in D2.3 - Social Experience and Gamification techniques for increasing attractiveness. The corresponding interface design details are explained in deliverable D3.2 – Situated and Attractive Information Exchange Techniques for Workers.
REFERENCES

AngularJS, https://angularjs.org/
Angular Material, https://material.angularjs.org/
Liferay, https://www.liferay.com/
LinkSmart, https://www.linksmart.eu/redmine
PuTTY, http://www.putty.org/
Python, https://www.python.org/
RESTful, http://restfulapi.net/