eCraft2Learn

Digital Fabrication and Maker Movement in Education
Making Computer – supported Artefacts from Scratch

Deliverable D4.4

The unified user interface - A software solution for 3D design, programming and making computer-supported artefacts (M24)

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 731345.

PROJECT DESCRIPTION

Acronym:	**eCraft2Learn**
Title:	Digital Fabrication and Maker Movement in Education: Making Computer-supported Artefacts from Scratch
Coordinator:	University of Eastern Finland
Reference:	731345
Type:	RIA
Program:	HORIZON 2020
Theme:	Technologies for Learning and Skills
Start:	01. January, 2017
Duration:	24 months
Website:	http://www.project.ecraft2learn.eu/
E-Mail:	office@ecraft2learn.eu

Consortium:

University of Eastern Finland, Finland, (UEF), Coordinator

Edumotiva, Greece (EDUMOTIVA)

Mälardalen University of Sweden, Sweden (MDH)

Zentrum für Soziale Innovation, Austria, (ZSI)

The University of Oxford, United Kingdom, (UOXF)

SYNYO GmbH, Austria, (SYNYO)

University of Padua, Italy, (UNIPD)

Technopolis City of Athens, Greece (TECHNOPOLIS)

Evothings, Sweden (EVOTHINGS)

Arduino, Sweden (ARD)

Ultimaker, United Kingdom (ULTIMAKER)

Linnaeus University, Sweden, (LNU)

DELIVERABLE DESCRIPTION

Number:	**D4.4**
Title:	**The unified user interface - A software solution for 3D design, programming and making computer-supported artefacts**
Lead beneficiary:	UOXF
Work package:	WP4
Dissemination level:	Public (PU)
Type	Other (O)
Due date:	31.12.2018
Submission date:	31/12.2018
Authors:	**Ken Kahn** [UOXF], **Afshin Ameri** [MDH]
Contributors:	All consortium partners
Reviewers:	**Christian Voigt** [ZSI]

Version Control

Version	Date	Person in charge (Organization)	Changes	Quality Assurance
1	08.12.2018	Ken Kahn (UOXF]	Creating final version	UOXF
2	11.12.2018	Ken Kahn (UOXF]	Integrated changes by Afshin Ameri [MDH]	UOXF
3	15.12.2018	Ken Kahn (UOXF]	Implemented suggestions by Calkin Montero [UEF]	UOXF
4	18.12.2018	Ken Kahn (UOXF]	Improvements by Baran Curuklu [MDH] and Ken Kahn [UOXF]	UOXF
5	20.12.2018	Ken Kahn (UOXF]	Incorporated Christian Voigt [ZSI] suggestions	Christian Voigt [ZSI]
6	26/12/2018	Ken Kahn (UOXF]	Incorporated suggestions from Calkin Suero Montero [UEF]	UOXF
7	29/12/2018	Ken Kahn (UOXF]	Incorporated suggestions from Calkin Montero and Tapani Toivonen [UEF]	UOXF
8	31/13/2018	Ken Kahn (UOXF]	Incorporated suggestions from Calkin Suero Montero [UEF]	UOXF

Acknowledgement: This project has received funding from the European Union's Horizon 2020 Research and Innovation Action under Grant Agreement No 731345.

TABLE OF CONTENTS

TABLE OF FIGURES

EXECUTIVE SUMMARY

This report describes the work that is carried out in order to design and implement the third version (version 3) of the digital platform, which plays a central role in the eCraft2Learn project. The digital platform consists of two parts: the unified user interface (UUI) and the teacher interface. The UUI addresses interaction with 3D technologies (modelling, visualisation, simulation, and printing) as well as DIY electronics (wiring, programming, debugging), thus provides feedback on design and design conditions of the project to the learner. The teacher interface is used by teachers to monitor and guide the learner. By separating the learner and teacher interfaces, we keep the learner interface simpler for users and are able to provide teachers with an interface with capabilities that are inappropriate for students. The description of the task as implanted against the DoA is given in the table below and it is further described in this document:

Task 4.4 content according to the DoA	Implementation Details
This task focuses on designing and implementing a unified user interface for educational extensions. This interface will address interaction with 3D technologies (modelling, visualisation, simulation, and printing) as well as DIY electronics (wiring, programming, debugging), thus provides feedback on design and design conditions of the project to the learner.	The UUI provides an interface to 23 different tools that cover the five pedagogic stages for the learner. These include tools for ideation, planning, 3D modelling, simulation, and printing, electronic design and simulation, programming, and sharing. It is integrated with the educational extension's learning analytics, AI programming, and feedback tools.
Thus, the work in this task consists of developing an interface that allows the users to interact will all equipment necessary through one interface only.	Via the UUI a learner can use Raspberry Pis, Arduinos, and connected electronics. 3D designs and slicing data can be created via the UUI and sent to 3D printers.
The same interface will also be used by the teachers to monitor and guide the learner. By designing one multi-purpose interface, for all users, the intention is to provide a holistic view of the process to the users as well as facilitate the involvement of the teachers in the learning process.	As described in detail in the section Relationship between final version and T4.4 we determined it was best to separate the learner and teacher interfaces. The teacher interface provides access to learning analytics as well as a means of customising the UUI's tools and resources.

1 INTRODUCTION

The UUI and teacher interface play a central role in the eCraft2Learn project. They are the prime tools for letting the learners and the coaches/teachers to access the technological solutions that are offered. The UUI interface is flexible enough to address needs in both formal and informal settings, although the rest of this document primarily describes issues associated with the former. The teacher interface provides access to the learning analytics, UUI customisation, calls for action, and approval of shared resources.

All of the digital tools that are required in the five stage craft- and project-based methodology of eCraft2Learn are accessed by the UUI. These steps include ideation, planning, creation, program and sharing.

The learning analytics section of the teacher interface is organized as a collection of several data mining tools found in a framework including (1) classification, (2) cluster analysis, (3) outlier detection, and finally (4) association rule learning. The learning analytics tools follows the eCraft2Learn “white box” methodology. The results of analysis are visually represented to the teacher. The teacher has the ability to adjust the models, which are the built classifiers and cluster analysers and hence, the classification and cluster analysis process itself produces domain knowledge as opposed to the results of the cluster analysis and classification. The learning analytics interface is very flexible allowing teachers to obtain insights without being overwhelmed by many tables, charts, and graphs.

eCraft2Learn Learning Methodology

eCraft2Learn is an ecosystem which is based on craft and project based teaching methods. This methodology includes five stages, which together guide the learner from an idea to developing and showcasing, through sharing, the final solution.

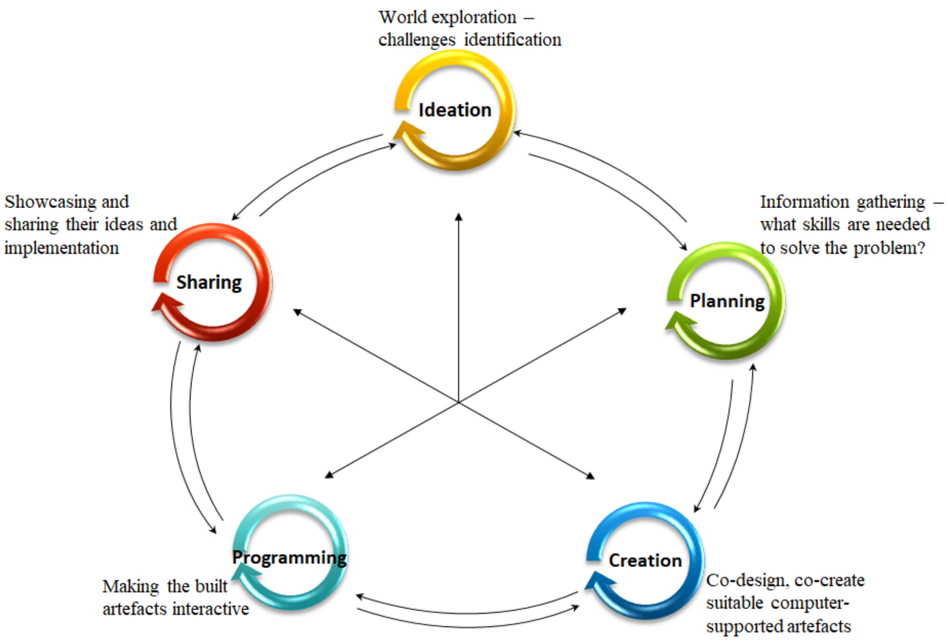

Figure 1 - eCraft2Learn craft and project-based methodology (the 5 stages)

- **Ideation:** The learners explore the world to find challenges. This exploration can be in the physical or virtual world (i.e. online communities).
- **Planning:** The learners collect information and material. They also start making project plans.
- **Creation:** Through a co-design and co-creation process, the learners start creating their solutions. This stage might involve many different technologies such as do-it-yourself (DIY) electronics, visualization, simulation and 3D printing.
- **Programming:** Learners add functionality to their crafted artefacts through high-level programming languages.
- **Sharing:** By sharing the solutions on online communities, the learners can learn from other projects, while receiving feedback from designers, engineers and programmers.

Each of the aforementioned stages requires a different set of tools and materials. Developing a central place (software) for managing these stages is the aim of the educational extension in eCraft2Learn. Such a solution will make it easier to manage the learners' co-creation project work and also can provide then additional tools to improve their projects even further.

The UUI plays a central role in this context since its aim is to allow the users to go between different tools smoothly. Another important feature of this interface is that it collects information about the activities of the users. This information is used in the learning analytics part. The learning analytics solution will collect data from technical parts of the eCraft2Learn framework, such as 3D printers and programming environment, as well as from learners' and teachers' interactions with the (UUI).

Furthermore, the UUI and the Learning Analytics (LA) system interface support the pedagogical processes of self-regulation and self-reflection by enabling the communication between teachers and students through the interfaces. In the LA system interface the teacher is able to track the groups progress and based on this the he/she can prompt the students to self-reflect via the Call4Action module. Moreover, personalised learning is also supported through the interfaces though the collection of Achievements. This mechanism is based on the co-evaluation of the students' progress as perceived by the students themselves as well as the teacher.

1.1. RELATIONSHIP BETWEEN FINAL VERSION AND T4.4

Task 4.4 content according to the DoW	Implementation Details
This task focuses on designing and implementing a unified user interface for educational extensions. This interface will address interaction with 3D technologies (modelling, visualisation, simulation, and printing) as well as DIY electronics (wiring, programming, debugging), thus provides feedback on design and design conditions of the project to the learner.	The UUI provides an interface to 23 different tools that cover the five pedagogic stages for the learner. These include tools for ideation, planning, 3D modelling, simulation, and printing, electronic design and simulation, programming, and sharing. It is integrated with the educational extension's learning analytics, AI programming, and feedback tools.
Thus, the work in this task consists of developing an interface that allows the users to interact will all equipment necessary through one interface only.	Via the UUI a learner can use Raspberry Pis, Arduinos, and connected electronics. 3D designs and slicing data can be created via the UUI and sent to 3D printers.
The same interface will also be used by the teachers to monitor and guide the learner. By designing one multi-purpose interface, for all users, the intention is to provide a holistic view of the process to the users as well as facilitate the involvement of the teachers in the learning process.	As described in detail below regarding the relationship between final version and T4.4 we determined it was best to separate the learner and teacher interfaces. The teacher interface provides access to learning analytics as well as a means of customising the UUI's tools and resources.

The interface as described in this document deviates from the task description only regarding the idea of a single interface for both learners and teachers. After the learner functionality was well-supported we began to design and build the teacher/guide functionality of the interface. While of course sometimes teachers will use the student-oriented functionality we determined that there is a good deal of functionality that is uniquely of value to teachers. The reports from learning analytics, the uploading of custom resources, the teacher part of the badging system, and the publication of student shared projects are only of value to teachers. Providing these as part of the same interface as that used by students would

- unnecessarily complicate the learner's experience
- give students access to inappropriate functionality
- combine disparate tasks into one interface where teachers are more productive if they can focus on the uniquely teacher tasks without the distraction of an interface that also supports learners.

Technically we could have easily merged the learner and teacher functionality but as we began to do so we discovered that keeping them separate had many more advantages than disadvantages.

In our reports, including this document, we use the term "eCraft2Learn digital platform" to refer to both the learner and teacher interfaces of the eCraft2Learn learning ecosystem.

2 Unified User Interface version 3 (Learner interface)

The UUI is a portal to a wide range of tools that covers the eCraft2Learn five pedagogic stages. The majority of these tools are integrated in that they appear as elements of the UUI that can be treated as sub-windows. The integration also includes data gathering for learning analytics and file sharing. The educational extension which hosts above mentioned tools, and its integration with the UUI is described in detailed in D4.3 and a user manual for its use and a video demo are included in D4.5. As can be seen below (Figure 2), the most evident feature is the clusters of set of tools for each of the five stages (Ideation/Imagine, Plan, Create, Program, Share) of the eCraft2Learn craft- and project-based learning methodology. As can be seen all tool are visible and available, at the same time the five stages of the methodology are evident.

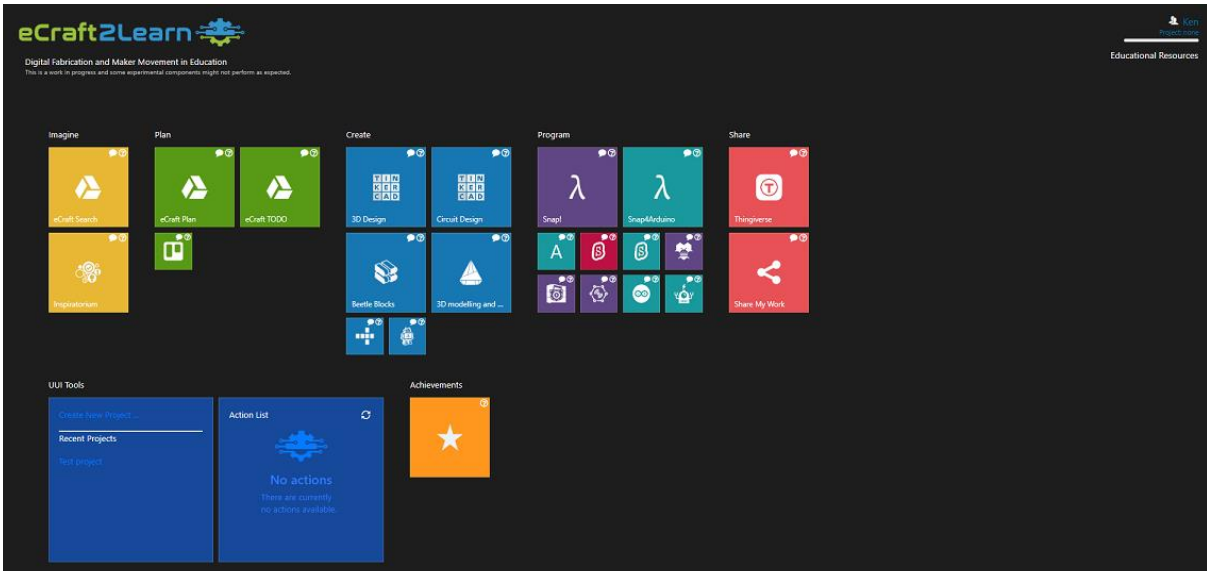

Figure 2 - The UUI is a 2D space. Accessible from https://ecraft2learn.github.io/uui

2.1. User interface approach and design

The term “unified” in Unified User Interface refer to the fact that the user, a learner, can access different tools for making though one single interface. Collecting vastly different tools, from different resources, in one single interface has been one of the objectives of the project. The motivation was to test, firstly, if this is possible to achieve, and still meet the pedagogical requirement of the project. Secondly, to investigate how missing parts, services, with respect to the pedagogical model could be compensated, through implementation by the partners. Thus, the development of the UUI and the tools in the educational extension were developed as an answer to the requirement that came from the pedagogical requirements, together with the outcomes of the pilot activities.

There are many ways of integrating tools into a unified interface resulting in different degrees of unification. For tools developed by a single entity it is possible for the look and feel of each tool to be consistent. They can share files and interoperate smoothly. Due to the huge development effort of doing this we pursued it only for a small number of tools for which open source or free alternatives were not suitable. An intermediate level of integration can be achieved by altering the source code of third-party tools when licenses permit. This introduces a difficult maintenance problem to keep the altered version up-to-date. We did this only in very few cases to integrate better with the learning analytics. Snap4Arduino is the best example where project files needed to be made available to the learning analytics subsystem. The approach we took for the remaining tools provide a weaker but useful integration by launching the tools within the UUI as “iframe” panels that can be manipulated. In addition, we integrated the help and feedback of all tools within the UUI. A few tools provided freely as third-party services do not even allow iframe panels and they need to be launched in a separate tab.

The Unified User Interface can be found at https://ecraft2learn.github.io/uui/. We decided to use the Metro style (https://en.wikipedia.org/wiki/Metro_(design_language)) familiar from Windows 10 and elsewhere. It is structured around the five pedagogic stages of eCraft2Learn, as well as taking into account other pedagogical processes that underline the apprach. Each tool provides both tooltips and a more detailed help page. We decided to highlight those tools that are tightly integrated and recommended by making their icons much bigger than other tools. When possible, tools are launched as sub-windows of the interface. These can be minimised, maximised, and closed. When there is no web-based version of a tool the Unified User Interface launches a sub-window describing with images and videos how to launch the desired tool. The UUI updates logs of user actions on one of our servers to facilitate learning analytics.

The UUI is hosted on github.io to facilitate collaboration and versioning. Github.com is the free open source code repository that is used in the project. The HTML, CSS, and JavaScript files are automatically copied from github.com to github.io where they are served as web pages. It is highly robust and reliable. Being free means the site will remain after the project and its funding has ended.

The UUI website only serves static files and hence requires no maintenance on our part. Updates and security is handled by Github. In order to collect data for learning analytics JavaScript code was added to make AJAX calls to a project server in Finland the stores the user logs in a database.

By relying only upon static web pages the UUI can easily be run using the local file system or local web server in situations where there isn’t reliable Internet access. The project also provides local copies of those web-based tools that are open source.

To implement the popular Metro style interface, the project group decided to build upon Metro UI CSS (https://metroui.org.ua/). It was developed with the advice of Microsoft (the original designers of this UI style) and includes general styles, grid, layouts, typography, over twenty components, and over three hundred built-in icons. Metro UI CSS uses the MIT open source licensing model.

2.2. INTERACTING WITH UNIFIED USER INTERFACE

2.2.1. INTERACTING WITH TILES

There are 3 different ways to interact with a tile: hovering the mouse over a tile, clicking a tile and clicking the question mark on top of the tile. Hovering a mouse over a tile shows a short description of that tile on it. This can be seen in Figure 3.

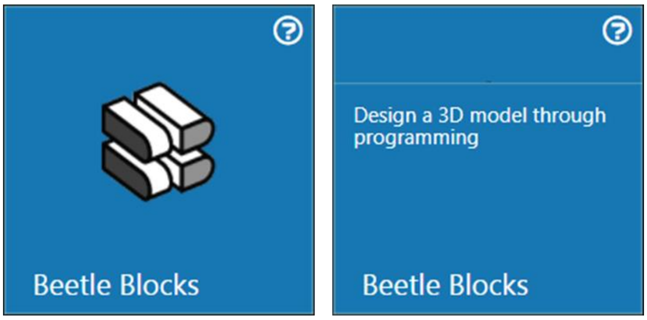

Figure 3 - The two states of a tile. The right one shows the tile while a mouse is hovering on it. This activates a short description text about that tool.

Clicking the question mark on top-right corner of a tile, opens a small dialog which explains that tool in more detail. The text might contain links that you can follow to learn more about a tool. As an example, shows the help dialog for the Snap! Tool on the UUI.

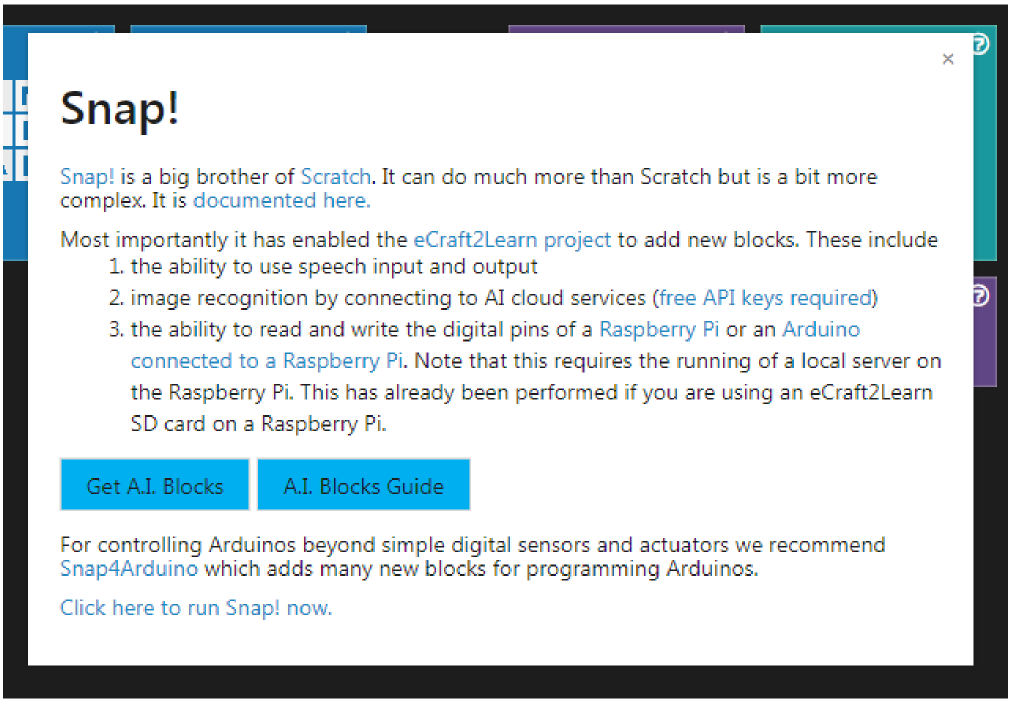

Figure 4 - The help dialog for the Snap! Tool. The help dialogs can be accessed by clicking on the question mark located on the top-right corner of each tile.

Clicking a tile will do one of the following:

- Opens the tool in a new window inside eCraft2Learn UUI (whenever technically feasible).

- Opens the tool in a new browser window/tab.
- Shows a short video and description on how to launch the tool from the main operating system.

2.2.2. Launching a Tool in a UUI Window

When technically feasible we launch tools as a subwindow of the UUI. The tools launched in this manner are eCraft Search, Inspiratorium, eCraft Plan, eCraft TODO, 3D Modelling, 3D Slash, TinkerCad 3D Design, TinkerCad Circuit Design, Snap!, Snap4Arduino, Share My Work, and Thingiverse. Launching a tool inside a UUI window allows you to access other tools directly from the UUI without the need to move between different operating system windows or browser tabs. Figure 5 shows TinkerCad launched in a UUI window.

It is possible to close, maximize and minimize the window, using the three command buttons on the top right corner of the window.

Maximizing a window: Click on the middle button on the top-right corner of the window. By using this option, you allow the tool window to use all the screen space possible. Clicking on this button again bring the window back to its original size.

Closing a window: Click on the X mark on the top right corner of the window. Remember that closing a window ends the session with that tool and the next time that you open that tool you may need to reload/reopen your project. If you want to keep your workspace open in the tool and temporarily use another tool, then use the minimize option.

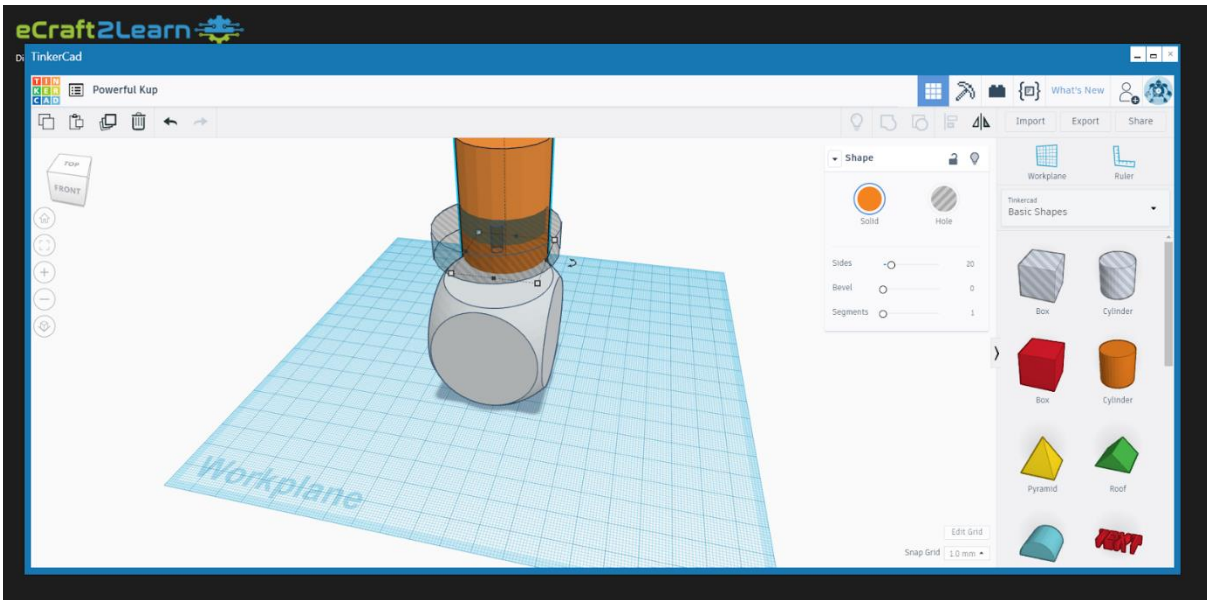

Figure 5 - TinkerCad running in a window inside eCraft2Learn environment

Minimizing a window: Click on the third button from right on the top-right corner of the window. This hides the window from your view, giving you access to full interface of the UUI.

The contents of the window and your project stay active and a small icon representing that tool is added to the active tools panel at the bottom of the page.

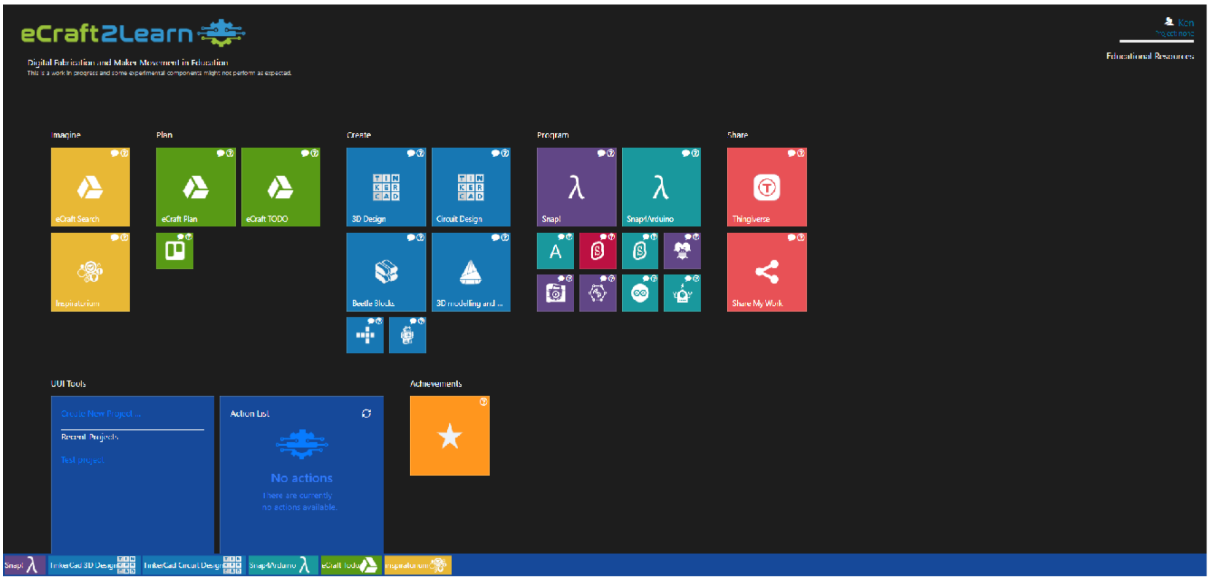

Figure 6 - The UUI with the Active Tools Panel visible.

The active tools panel contains a list of all the minimized tools for easy access by the user. Please note that it is possible to launch a tool several times (for example if the user wants to open two different files in the same tool).

2.2.3. User Login to UUI

The login page is automatically shown on the first access to the interface. Users need to input their username (often assigned by the teacher). One needs to always use the same username shown the login page. Often the username is the project team's name. During the pilots the user needs to choose the pilot site from a list. Alternatively, a session id can be generated in the teacher interface and entered here. The session id or pilot site name are important for scoping the learning analytics.

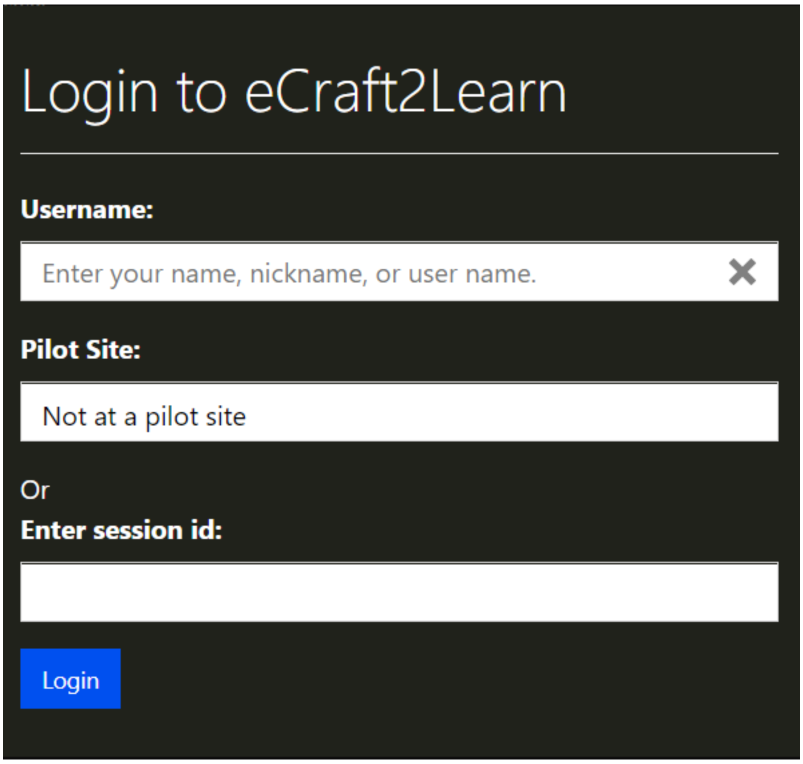

Figure 7 - The login page

2.2.4. Teacher Notes

In the login page make sure that the students understand the importance of using the same user name for subsequent logins. This helps the data analytics and debugging systems to give a better feedback to the user.

User authentication

The login page does not ask for passwords. For the pilots this was considered a burden to both the users and the developers that gained little. Regarding privacy and data protection, we covered the pilots from all perspectives (e.g., ethical approval from the university board, parents' consent, teachers' consent, students' assent). Regarding dissemination of the digital platform to schools and informal learning centres, they have a choice of running the data collection and learning analytics on their own server or using the eCraft2Learn ecosystem without learning analytics. Without learning analytics no data is gathered and hence no user authentication is needed. Schools that set up learning analytics on their own servers can easily add the local password protection that teachers currently use for other school services. Note that the sharing tool enables anyone to upload files but they are available only after approved by a teacher.

2.3. Changes to the Unified User Interface since Version 2

Many changes influenced by participatory design with teachers and students have been made since version 2 (M18):

1. Integration of the project management system with feasible UUI tools allows the tools to be launched with up-to-date state from the active project.
2. Further customisation of the interface allows the teachers to deactivate specific tools in the UUI for a given session.
3. Integration of the sharing tool allows the students to share their work with other eCraft2Learn sites.
4. Badging system is fully integrated with the UUI and acts as a self-evaluation tool.
5. The new eCraft2Learn 3D design and slicing tool provides 3D design and physics simulation as an alternative to TinkerCad 3D design.
6. Mobile version of the UUI is now fully functional and provides a subset of UUI tools which are suited for mobile devices.

2.4. Tools for Ideation

In D3.1 (Development of Personalised, Craft- and Project-based Learning) the ideation stage is described as follows (note that the ideation stage is called Imagine in the UUI, since it was assumed that the latter term would be easier to understand by the learners):

> "Ideation through world exploration: In order to explore the kinds of challenges that students or others face in their daily lives, a student could explore the world physically (e.g. taking pictures, exploring situations outside the classroom, newspapers, etc.) or virtually (e.g. through online support community discussion, online news, documents, local news/websites) and then decide what their challenge/problem will be. This process is also guided by the STEAM coach."

The initial version of the UUI and EES included Google Drive and Coggle as Ideation tools. The rationale for using Google Drive was that it can be used as storage space for the ideation phase that happens in the physical environment. For example, when outside the classroom, a learner might face a situation that inspires them and they take a photo of it. Google Drive can then be used to share this with others and start ideation.

The second tool that was integrated to the first version of UUI and EES was Coggle. Coggle is a web application that allows the users to create mind maps and ideas collaboratively. Creating mind maps is done by adding text, images and links to the mind maps. Any group member or the STEAM coach can add, remove or modify the document or comment on specific parts of it. This way the group can collaboratively improve their ideas and prepare them for development.

During initial eCraft2Learn pilots it was observed that the learners prefer using online search and finding related projects done by online communities as a form of ideation instead of using tools like Coggle. It was also observed that Google Drive is seldom used as a storage space for sharing ideas, as the learners normally prefer using other means of communication.

The lessons learned from the pilots, allowed eCraft2Learn to focus more on the tools that provide support for ideation in a virtual environment. Also, because both Google Drive and Coggle required extra logins apart from eCraft2Learn environment, they both were removed in the final version.

2.4.1 ECRAFT SEARCH

The eCraft Search tool can be used to search for ideas during the making projects. The tool introduces a search field where the relevant keywords can be entered, similar to any other text-based search tool. Afterwards, the tool makes queries to image and video sharing APIs and Wikipedia. Finally, the results are shown in the interface. For the video sharing, the tool queries YouTube and shows the most relevant educational and robotics related results. Also, for the image search API, which is Flickr, the keywords that the tool uses are educational and robotics related. For instance, if a student uses a search term 'solar system', then the API queries would search every solar system video and images that are tagged as 'robotics' or 'educational'. For Wikipedia, no additional tags are used, and thus, the tool queries every article containing the term 'solar system'.

Furthermore, the tool performs safe searches in order to filter out content not suitable for the children. Also, from the teacher's interface, it is possible to add more fixed keywords for the sessions. Adding search term that are not visible for the learners has a purpose. In required for the problem at hand, the teacher/coach can direct the search, or even eliminate certain results. this possibility may also be relevant if the learners are on different levels, or age groups. This function is part of the teacher's portfolio, thus whether it should be used or not is something that the teacher should decide (Figure 8).

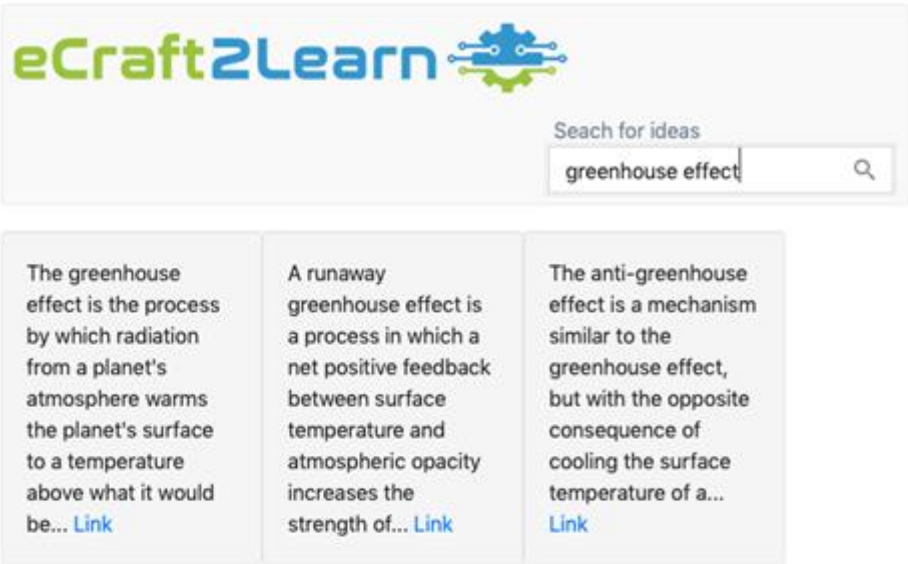

Figure 8 - The Interface of the Search Tool.

2.4.2 INSPIRATORIUM

The Inspiratorium provides learners and facilitators with a visual search interface to browse projects published on *instructables.com*. While the native instructables.com search is text-based and filters the project titles and descriptions, the Inspiratorium search displays the 100 most popular keywords as colourful bubbles. As a user clicks on bubbles, the search results narrow down to only those projects that are assigned all selected keywords. The search results are presented as a project list, ordered by popularity (Voigt, Mair et al. 2018). Selected keywords are coloured orange, possible additional keywords are coloured yellow, and keywords that were not assigned to any of the projects that match the previous selection are coloured grey (Figure 9). This immediate visual response has a playful element and stimulates users' creativity by showing unexpected combinations (e.g. "music, jewellery, furniture") and speeds up the exploration of Instructables projects.

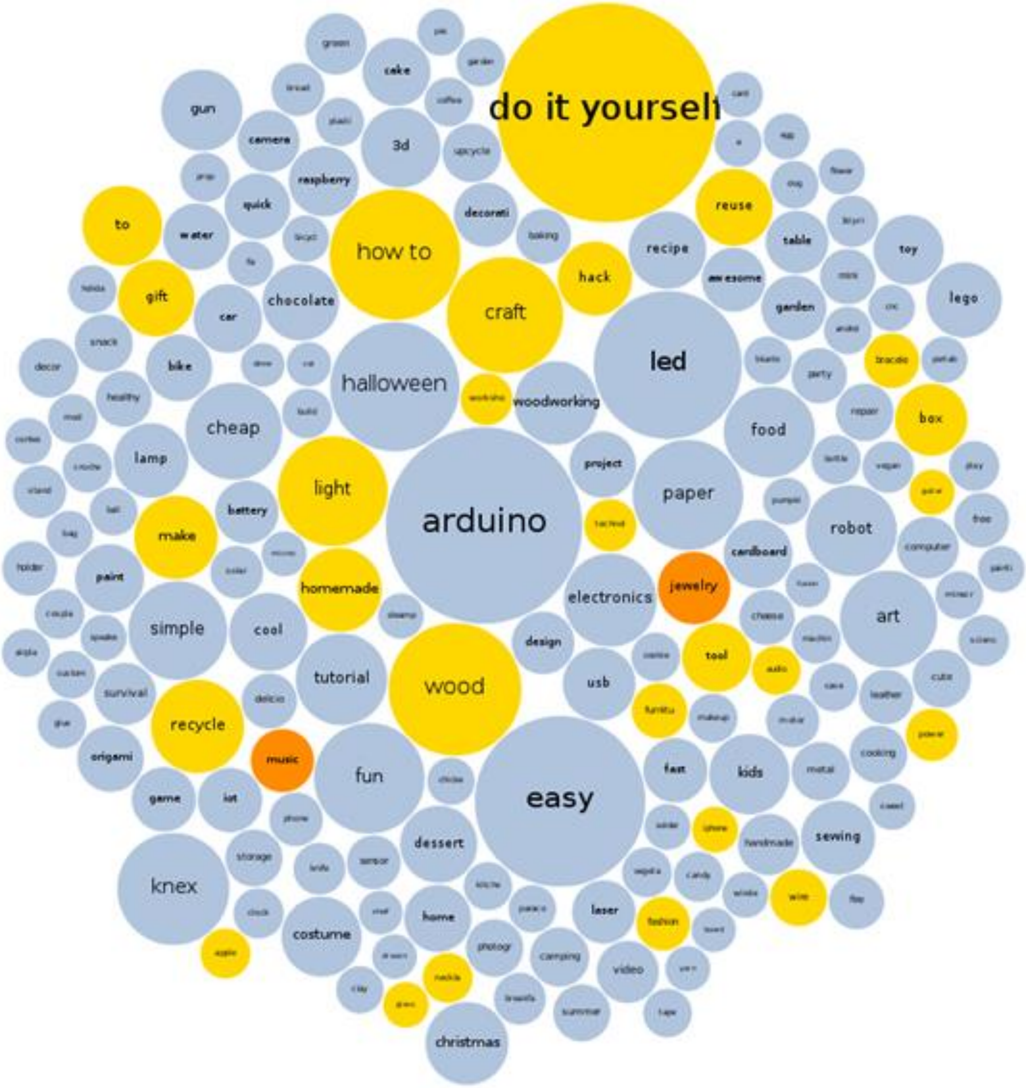

Figure 9 - Screenshot of a sample Inspiratorium search query (find projects with keywords "music" and "jewellery"), selected keywords are coloured orange, possible additional keywords are coloured yellow.

2.5 Tools for Planning

The planning tool should allow the learners to define a timeline, collect material (digital and non-digital) and assign roles to group members. Initially this was achieved by using Trello, a free online tool for simple project management in groups. Throughout the pilots it was observed that Trello, although a viable tool, cannot be the main planning tool for eCraft2Learn due to the following considerations:

- During the pilots, it was observed that the learners and teachers avoid using Trello as it requires a separate account and login.
- It was also observed that learners prefer to use pen and paper to draw their plans for the project instead of defining it as a set of tasks.

Based on the aforementioned observations, the eCraft2Learn team implemented "eCraft2 Plan" and "eCraft2 TODO" tools which together can provide the planning functionalities that the learners and teachers were expecting. It should also be mentioned that Trello is still available through the UUI, as some educational organizations already use it as a planning tool. For scenarios running on such organizations, Trello might be a better choice as the learners and teachers are familiar with the tool, have their logins already set up and use it frequently. In the following sections eCraft2Learn planning tools (eCraft Plan and eCraft TODO) as well as Trello are explained.

2.5.1. eCraft Plan

eCraft Plan is a tool developed for sketching plans in project making (Figure 10). The tool offers options to draw, write text, and upload images. Thus, this tools allows mixture of textual and visual information representation for planning and design of future projects. The tool uses HTML5 canvas elements where the layers of the drawings will be inserted dynamically. The resulting sketches can be saved for sharing or further use through the Educational Extension System File Management system.

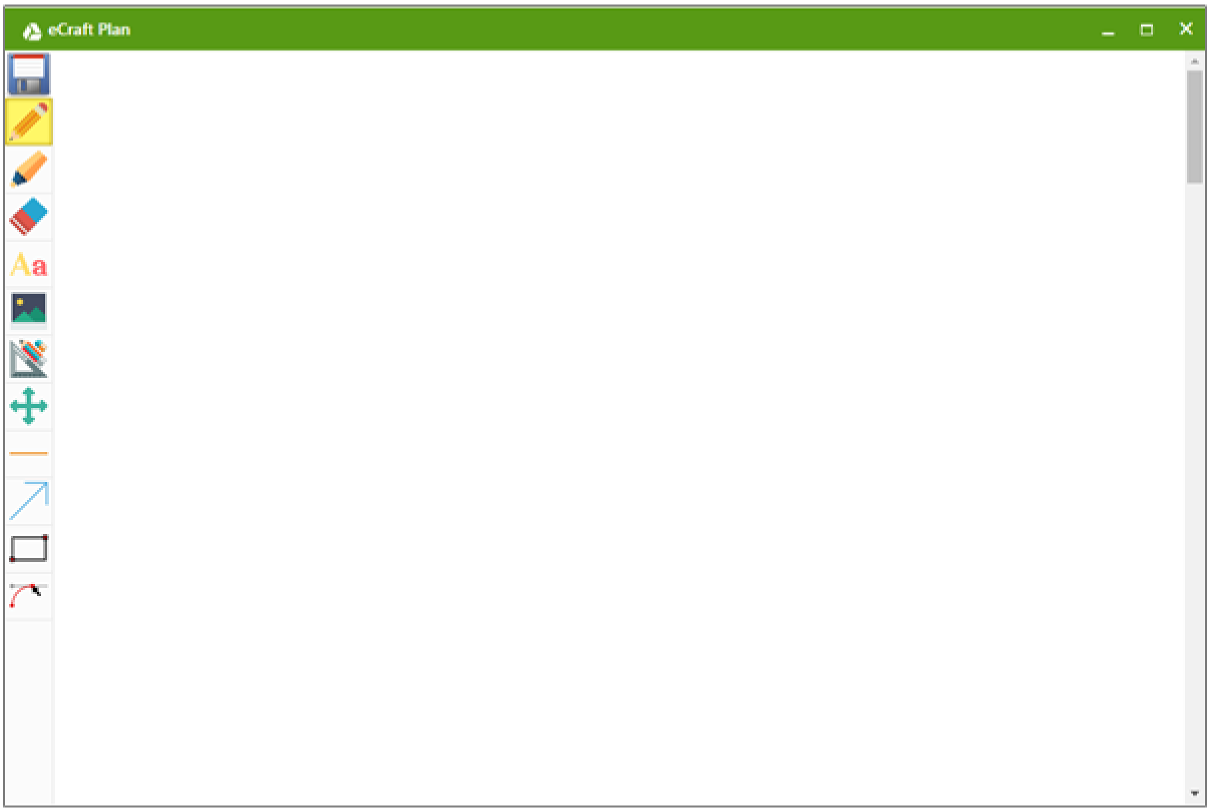

Figure 10 - eCraft Plan's Interface.

2.5.2. ECRAFT TODO

In eCraft TODO (Figure 11), it is possible to plan and schedule tasks within the group. The user can enter the name of the tasks, the description of a task, to whom the task is assigned and the due date. The tasks can have three different labels: pending, in progress and completed and the tasks can be manipulated by drag & drop between the labels.

The user can save (using the “save” button) the process of the group through the File Management system and thus, the tasks and the status of the tasks will be loaded from the server whenever the tool is opened.

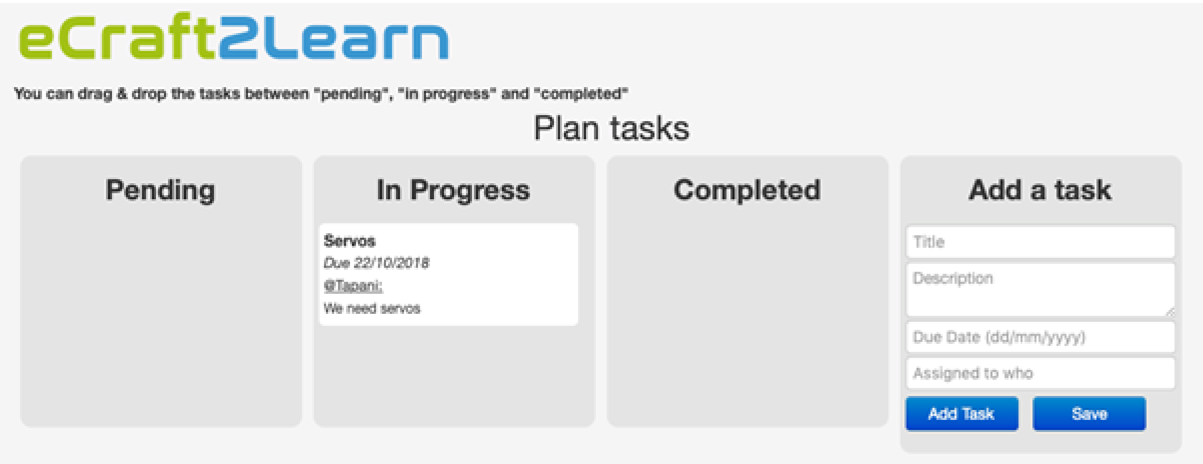

Figure 11 - eCraft TODO's User Interface.

2.5.3. Trello

Trello is an easy to use web application designed for collaborative project management. In Trello, the work is divided into boards, and each board consists of several lists (or cards) that represent different stages of the work or different tasks. Each task can have a deadline, can be assigned to a different group member and can also have different material (e.g. from ideation stage) attached to it. Trello also has direct communication with Google Drive.

Trello makes it easy for the teachers to supervise the project as well. Teachers can be added to the projects as group members and can comment on different aspects of the plan. They can even subscribe to the projects and be notified of any changes as they occur (Figure 12).

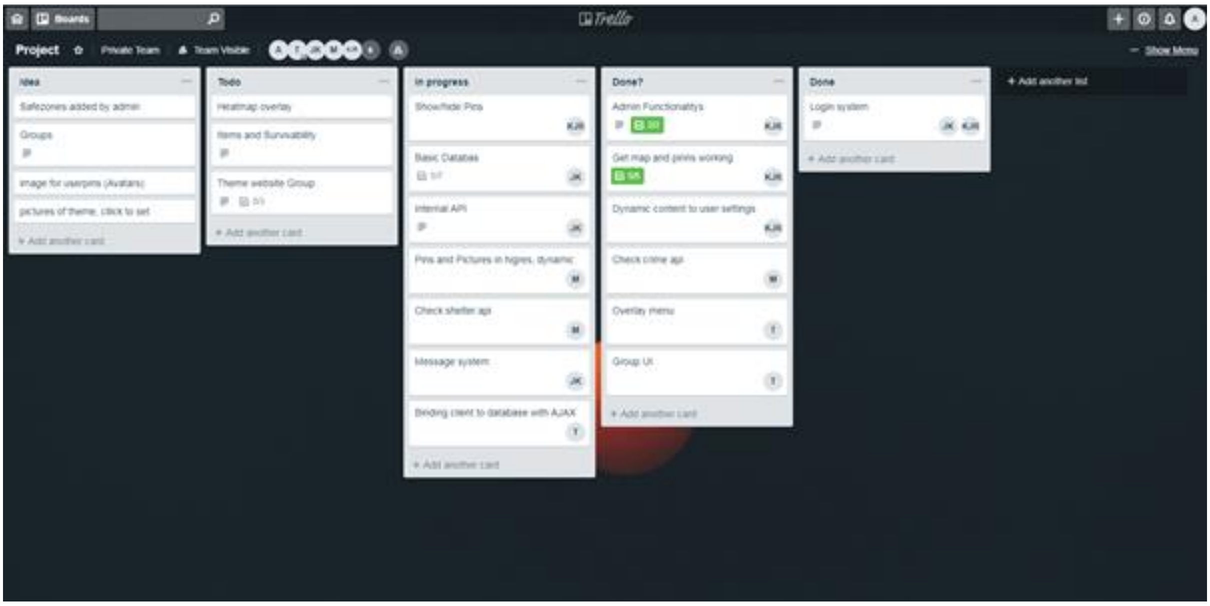

Figure 12 - An example of a group plan on Trello. Note that for each task, the initials for group members responsible for that task is also visible (grey coloured circles bottom-right of each task).

2.6. 3D Modelling and Printing Tools

2.6.1. Tinkercad Design

TinkerCad is a free, web-based 3D modelling and printing web application from Autodesk which is aimed at young learners and makers. It has a shape-based 3D modelling interface, where learners can create new shapes by manipulating basic geometrical objects, performing logical operations on them (conjunction / disjunction) and/or using pre-built 3D objects (eg. Arduino cases). TinkerCad can also generate STL files for 3D printing from user designs.

The software allows for collaborative work on 3D designs and the learners can invite other group members to join them in tinkering the 3D models. Sharing 3D designs is also easy when using TinkerCad, as it can directly share the designs on “Thingiverse” (Figure 13).

Tinkercad is chosen as one of the main 3D design tools in eCraft2Learn because of its wide use, simple interface and community support. It also provides DIY hardware simulation and debugging in the form of Tinkercad Circuits (Section 2.4.1).

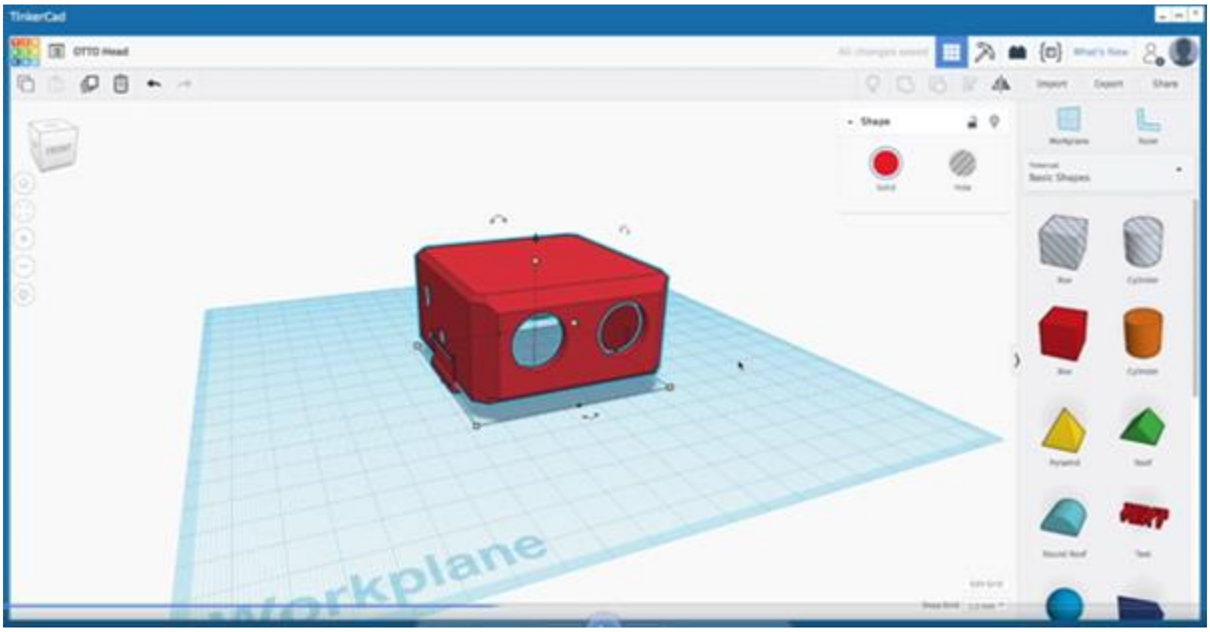

Figure 13 - TinkerCad's Interface

2.6.2. Beetle Blocks

Beetle Blocks is another 3D Modelling tool which is presented in eCraft2Learn’s UUI. It is a graphical block-based programming environment for 3D design and fabrication. It allows students to program a design in 3D using Snap! programming blocks. The programming approach to 3D design allows the learners not get immediate visual feedback on their work which is turn helps them with understanding both 3D modelling and programming from a different perspective.

A Beetle Blocks program controls an extruding head that can have different shapes (e.g. sphere, cube, etc.) and can be moved around by the program, the learners can also start/stop extruding, change the shape and colour of the extruder and employ user inputs to create different models (Figure 14).

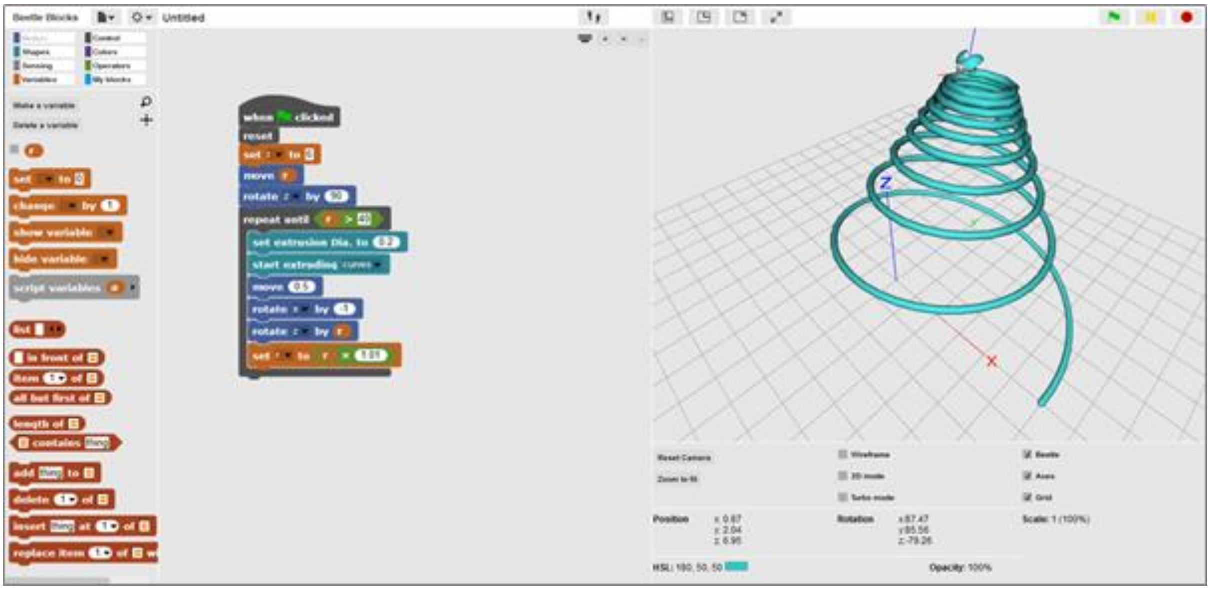

Figure 14 - A Beetle Blocks program and its generated 3D model.

2.6.3 CURA

Cura is the slicing software developed by Ultimaker. Since it does not have a web-based version, it is not possible to integrate it directly to the UUI. In order to use Cura in eCraft2Learn classrooms, Cura is installed on a separate computer and the students can access it using remote desktop access through VNC.

2.6.4 ECRAFT 3D

Within the UUI, it is possible to create printable web-based 3D models through eCraft2Learn's 3D modelling and slicing app. The tool is based on ThreeJS framework that uses WebGL to create 3D graphics. This means that in order to use eCraft 3D Tool, WebGL extensions should be enabled in the browser.

The tool allows users to add basic 3D shapes and modify them (Figure 15). The possible modifications include modifying the shape, the size and the rotation of the shape. The tool allows users to generate STL and GCODE files, which are supported by the 3D printers. The slicing of a GCODE files can be done in the browser or on the server, if the host computer is not powerful enough to slice the model. Later, the generated GCODE file can be imported to the 3D printer through a USB stick or an SD card.

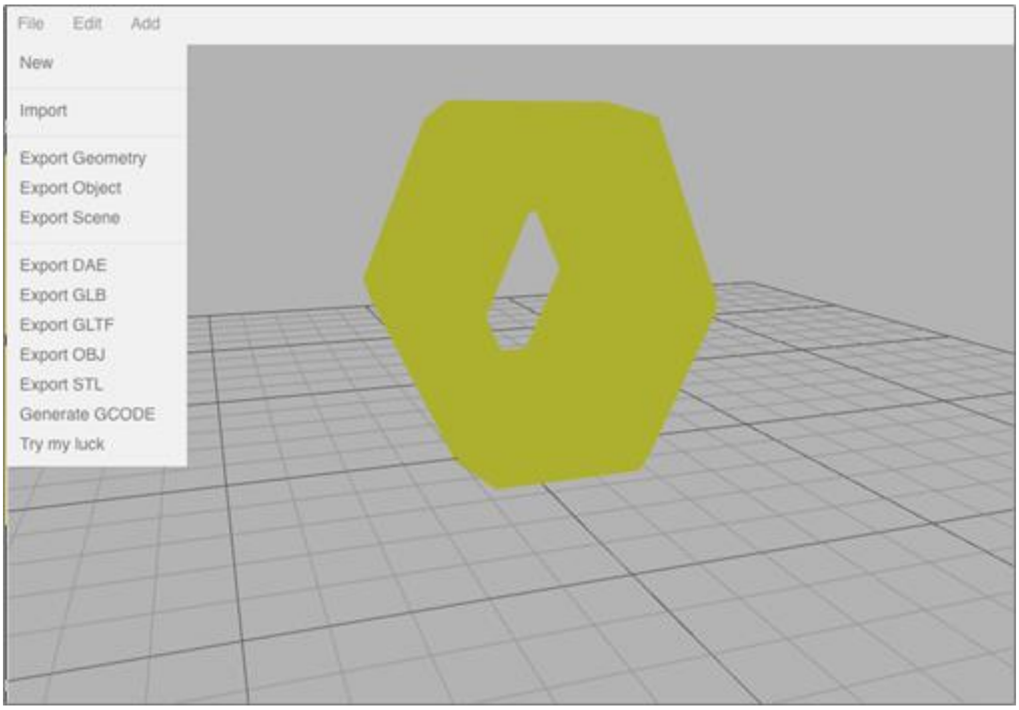

Figure 15 - 3D Modelling Tool's Interface.

2.7. Hardware Simulation and Debugging

2.7.1. Tinkercad Circuits

Designing DIY electronics and simulating the designs can be done using TinkerCad. “Circuits” is a part of the TinkerCad application (also known as “TinkerCad Circuits”) that allows the learners to design DIY electronics and simulate the behaviour of their design. This application has built-in support for Arduino UNO R3 microcontrollers and has a long list of different electronic components including different type of sensors, outputs and actuators. TinkerCad Circuits can also generate a “component list” when the design is finalized.

Similar to TinkerCad 3D it is possible to share the designs with classmates and teachers, although the sharing platform is not Thingiverse and is based on sharing direct links to invitees. Those who have the access link can collaborate in the design process by making changes to the design. Group members can easily share the links to circuit designs and 3D models on eCraft TODO and assign tasks to them.

“Circuits” also includes a code editor for developing in native Arduino language. The editor includes many different libraries and a debugger (Figure 16). The debugger provides tools to set up breakpoints in the code and watch the variables during simulation.

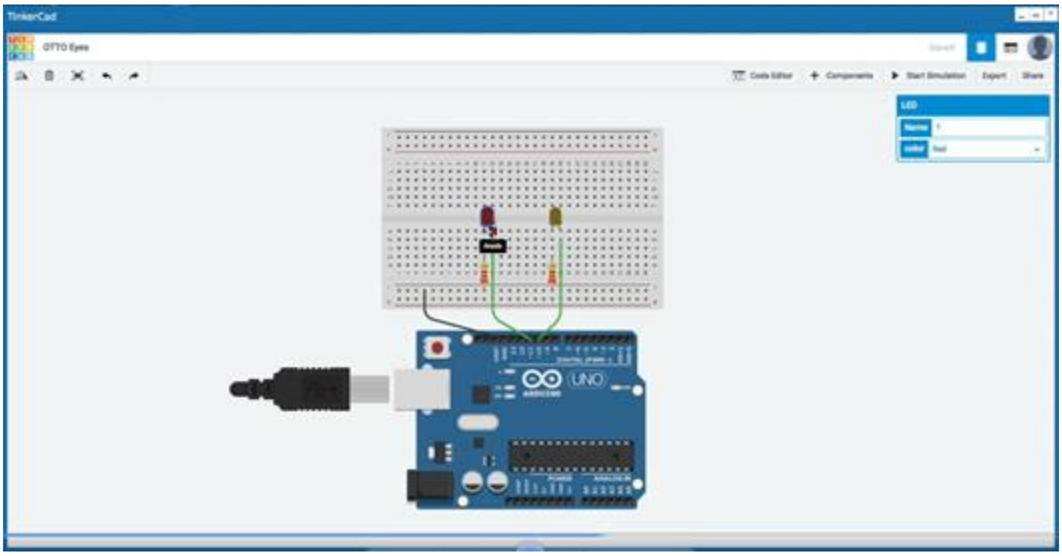

Figure 16 - The User Interface of TinkerCad Circuits.

2.8. ACHIEVEMENTS (THE BADGING SYSTEM)

Self-assessment and self-reflection may play an important part in the learning process. In order to provide the learners and teachers with tools to support this process, an approach to foster personalised learning through the use of badges during the eCraft2Learn activities is implemented. The aim is to provide a suitable solution to the first two challenges listed in the paper of (Miliband 2006), i.e.:

- Knowing the strengths and weaknesses of individual students.
- Develop competences and confidence in learner through teaching and learning strategies based on individual needs.

The badge system does not only aim at assisting the students with understanding their strengths and weaknesses, but also developing their confidence during the learning experience. Furthermore, the aim is also to help the learners with developing the capability to actively reflect on their learning process. This cannot be achieved if badges are used in the usual way, i.e. the teacher directly assigns the badges to the students on the basis of the outcomes of their activities and results during the implementation of their projects. Going beyond this, the badges are used within a self-assessment framework, where the students are requested to provide a self-evaluation of their own skills that they have developed during the practical activities. The self-assessment is composed of a score (in a scale from 1 to 5) and a textual justification for the score. The teacher reviews this self-evaluation and if necessary revises it, explaining to the students the reasons for the modifications. The teacher evaluations are converted into graphical scores in the form of badges.

Personalised learning has been fostered by a number of pedagogical approaches as well as tool. For instance, (McLoughlin and Lee 2010) discuss the feasibility of applying the social web tools (web 2.0) to foster personalised learning and self-regulation, through social media tools, such as Twitter™, urging for a pedagogical change to facilitate the integration of such social software tools. (Peirce, Conlan et al. 2008) propose an adaptive educational game architecture to foster personalised learning through gaming, avoiding the one-size-fits-all approach present in many educational games. Within the gaming approach, digital badges have also been introduced as a tool for boosting student's motivation, providing recognition of one's credentials as well as evidencing achievement (Gibson, Ostashewski et al. 2015). With this in focus, we propose a badge system that is based on the students' self-assessment.

In order to actively observe their own learning progression and, at the same time, to receive a sort of immediate recognition and gratification of their efforts, the eCraft2Learn ecosystem offers integrated evaluation and badge system based on students' self-assessment. The design of the badge system followed the same principle of personalisation aiming at being intuitive to use.

The badge system contains a self-assessment mechanism through which the students can evaluate their progress. Taking into consideration the suggestions of key teacher informants, we identified six categories for self-assessment, corresponding to the various aspects of a digital fabrication project development: building electronic circuits, 3D design, programming, presentation of the project outcomes, quality of the teamwork and originality. Parameters provided for each of these categories guide the students to achieve a reasoned and objective evaluation of their work.

When the teacher submits the created activity (or a request for self-reflection) (see Figures 17 and 18), it is available for the students. The teacher's request for self-evaluation shows in the Achievements panel in the UUI for the students to fill in. The teacher bootstraps the process always. The goal is the numeric value for each category "Programming", "Collaboration", etc. The students give the numeric value to themselves and the teacher can change it if he or she disagrees. Then the achievements are calculated from the teacher's actual numeric value.

The system provides default parameters to use for self-evaluation, however the teachers are able to provide their own parameters as well. Ideally, students and teachers should decide and set the parameters together, thus sharing the responsibility to represent the most relevant and meaningful aspects of the learning process. Based on the information gathered from teachers who were involved in the co-design process of developing the Badge System, we learnt that flexibility was needed in terms of allowing the teachers to modify the categories and their parameters in order to tune the evaluation to the aspects that they considered relevant to be self-reflected and self-evaluated by the students.

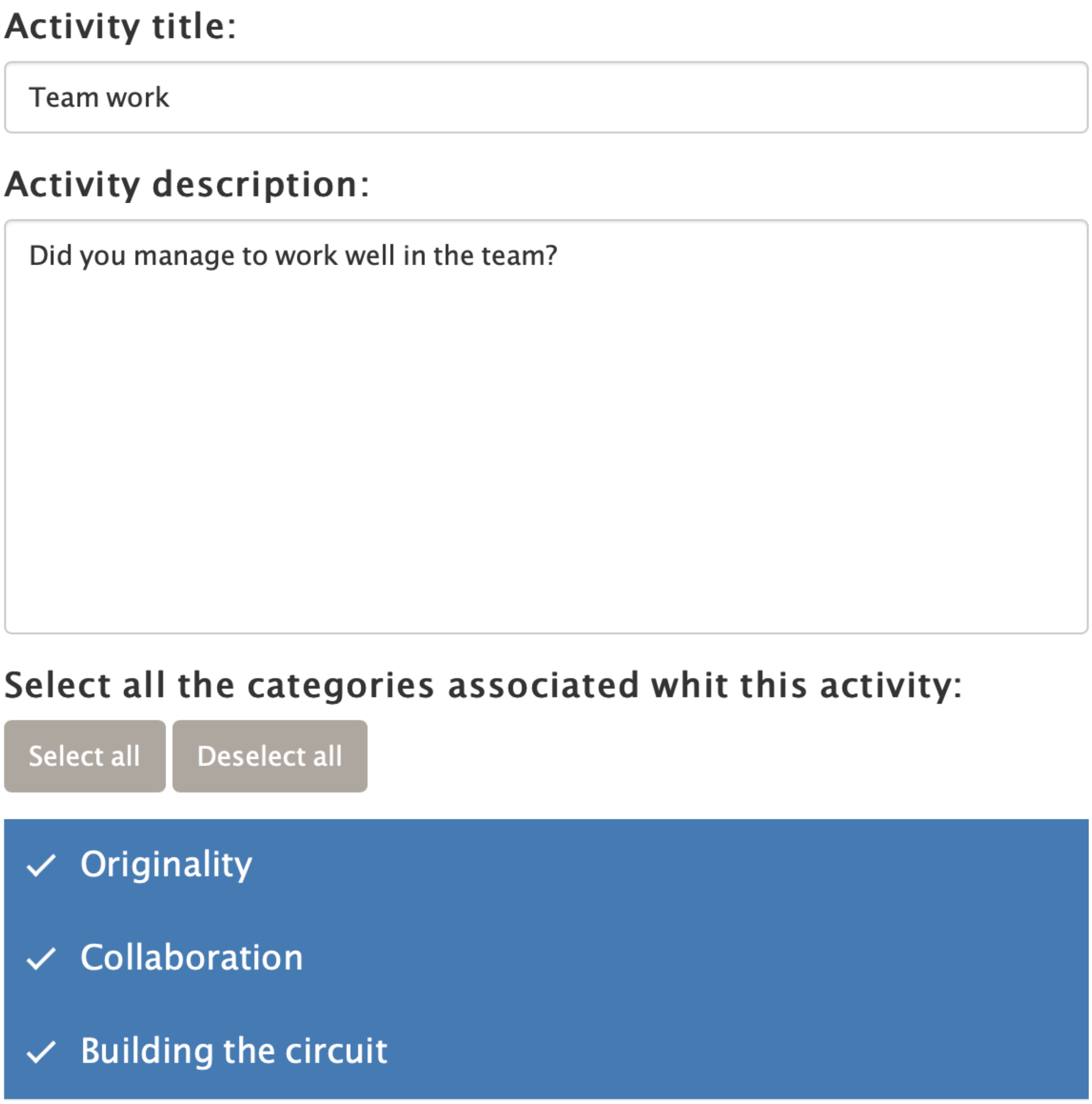

Figure 17 - The teacher interface for initiating self-assessment

The self-assessment system requires that the students first think about which parameters have been achieved. Later, they are asked to assign a score between 1 and 5 alongside a brief justification for their score to each of the six categories. In addition, after each activity the students are asked what the new subjects that they have learnt are, and what they found too difficult or they didn't completely understand. For this reason, even if reference parameters are available, the score that the students assign to their work is not necessarily tied to their actual achievements. Thus, the artefact created by the students in their project might not work perfectly, however the student could anyway self-assess a good mark, since they have learnt other important skills. The important aspect is that students provide justification to their considerations. For this purpose, the interface has text boxes where students should summarise the rationale behind the numerical scores that they assign to their work.

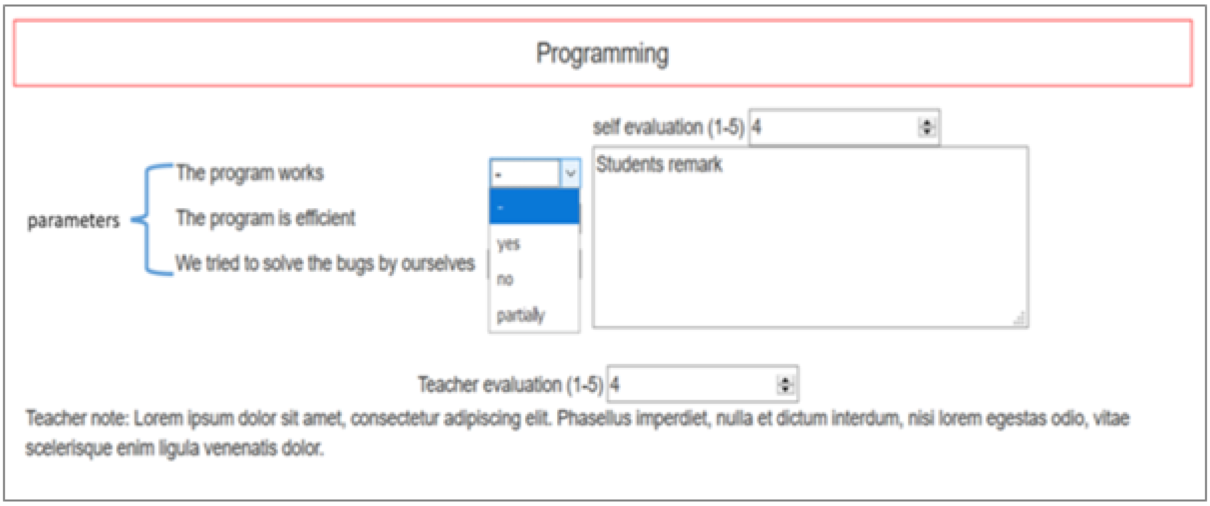

Figure 18 - The scores are confirmed by the teacher.

The information obtained from these rubrics is precious to the teacher, thus on the one hand it delivers data on the efficiency of the self-assessment process, and on the other hand it allows the teacher to judge if the learning activity was properly calibrated, offering a comprehensive vision of the strong and weak points of their students. The teacher can confirm a student's produced score or assign a different one in the self-evaluation interface, provided that the teacher justifies the change. The teacher modifications and justifications are visible to the students. This transparent evaluation from the teacher and this cyclic process of "self-evaluation ↔ evaluation ↔ assessment ↔ assessment of the self-evaluation" between the teacher and the students are key factors to develop the critical thinking of the students about their learning and the effectiveness of their learning strategies. Nevertheless, this cyclic process, the cognition and metacognition involved in it could result in overburdening the learning process. Therefore, in order to support the learners and keep them engaged in the learning process, we rely on gamification exploiting the quest for more badges and for "more stars" in each badge as a motivating and compelling feature.

The use of game design elements in a non-game context is called gamification (Deterding, Dixon et al. 2011). Since 2013, there has been a wide consensus on the gamification practice in the educational environment, especially at the higher degrees of education. Badges are an example of game mechanics, along with points, levels, progress bars, leader boards, virtual currency, and avatars. As several studies report, the use of these mechanics have a great potential in increasing students' engagement and in motivating them to learn and train new skills (Barata, Gama et al. 2013, Dicheva, Dichev et al. 2015). In particular, badges are proved to be effective in triggering competitive motivation (Pirker, Riffnaller-Schiefer et al. 2014), in improving learners' participation (Dominguez, Saenz-de-Navarrete et al. 2013) and in enhancing learning, time management and carefulness (Hakulinen and Auvinen 2014).

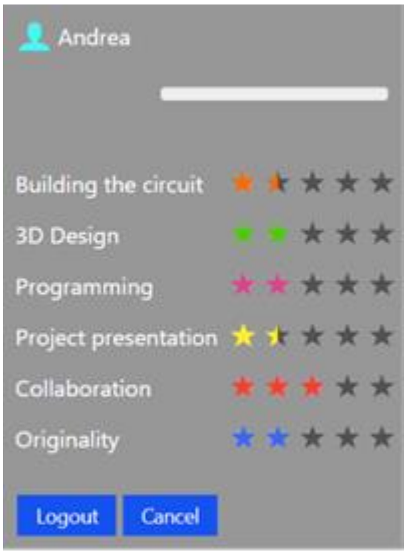

Figure 19 - The confirmed scores are displayed in the UUI as graphical badges.

In the eCraft2Learn digital platform of the learning ecosystem, the self-evaluation interface that we are presenting as a gamification process has been embodied into a set of badges. Indeed, the scores assigned by the teacher are elaborated by the eCraft2Learn system and converted into badges; they can also refer to the learning analytics system (Toivonen et al., 2018) in order to provide a more objective evaluation. Badges give students an immediate and easy to understand feedback about their progress in each category (Figure 19).

The badge system has twofold goal. On the one hand, it shifts the attention of the students from the practical activities they have to carry on to complete the project, to the skills they are supposed to learn during this activity. This is achieved by the careful choice of the six categories for self-assessment which were defined with the help of teachers involved in the eCraft2Learn's co-design process. Each of these six categories visually prompt the students to actively reflect on the fundamental skill they are supposed to assimilate abstracting their thinking from the technical/practical problems at hand. As one can notice, the six categories are divided into technical skills specific to digital fabrication and making activities (such as circuit building or 3D design) and the cross-skills to be acquired (such as presentation capabilities and quality of the teamwork).

On the other hand, the badge system motivates the students to complete their digital fabrication projects and the practical activities associated with the project. The more activities the students complete (and more carefully they implement them), the more badges they will earn. Moreover, in order to earn more badges, they have to finish up their work and critically revise and evaluate it, speculating on the achievements of their own learning. This process aims at fostering a reasoning on the outcomes of their learning. Additionally, the students are encouraged to develop (and express) a fair self-evaluation, because the teacher is evaluating both the technical achievements and the self-evaluation capabilities of the students.

2.9. File Management System

Many of the tools which are integrated into the UUI offer the option to save the learners' work into files. For a given project work there are several different tools to use where each of them can save the file locally or on the cloud. Since these tools do not have direct communication with each other, keeping the relationships between these file is a cumbersome task for the learners. Also, in the case of cloud storage, each tool might use its own cloud service which might require a separate login and will be unavailable to the learner in the case of no internet availability. Therefore, eCraft2Learn provides a "File and Project Management" system (FPM) that is fully integrated with most of the UUI tools. The requirements for this system are as follows:

- Provide a single cloud service for storing files related to UUI tools.
- Store relationship between a file, its related project and the tool which is used to create it.
- Provide a user interface in the UUI which allows simple migration between different projects.
- Should be lightweight so it can run on a local server.

The last requirement is designed to allow for running the eCraft2Learn platform locally, which will help in cases where internet connection is slow or not available.

Storing the relationship between files, user and projects allows the FPM to act as a filter that provides the integrated tools with the files which are only related to the specific user, project and tool (Figure 20). The files are stored in the web server which runs the FPM service which will be accessible through tools' cloud save/load options.

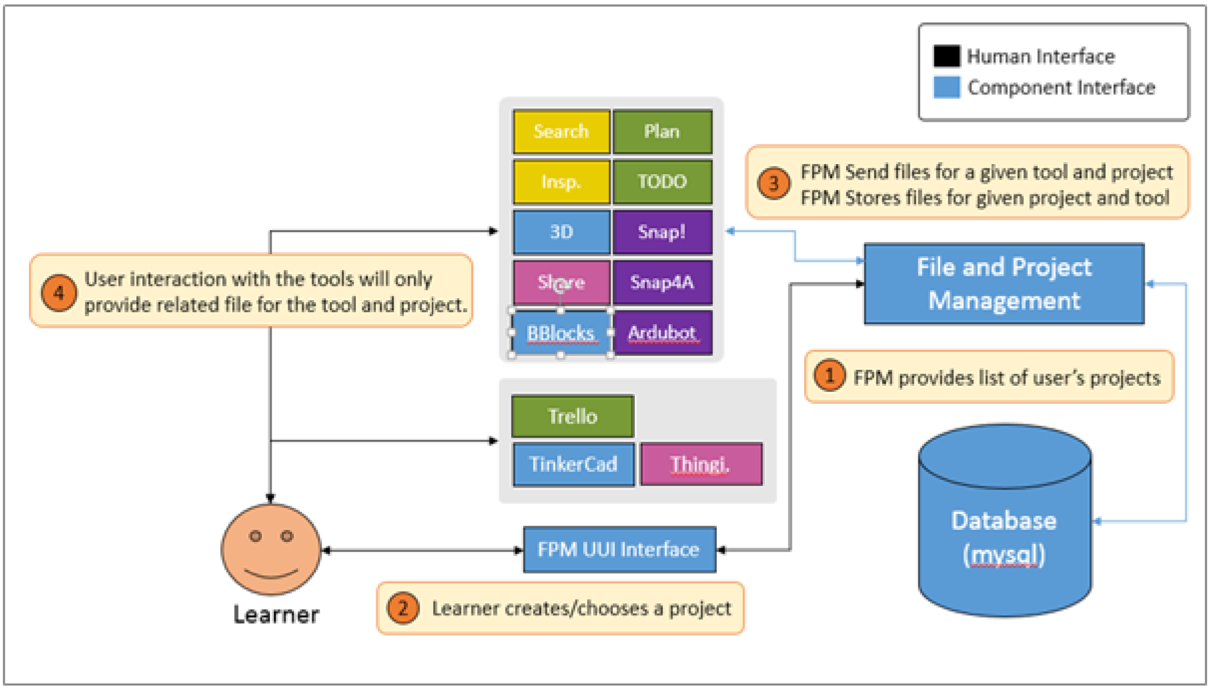

Figure 20 - Interfaces between FPM, UUI and the user.

2.9.1. Project Management Tile in the UUI

The project tile in the UUI (Figure 21) provides the learners with a link to create a new project alongside a list of previous projects for the group. This information is retrieved from FPM upon logging in to the UUI. Users can simple click on "Create New Project" and provide a project name in a new dialog (Figure 22) or click on one of the previous projects to activate it.

Figure 21 - Project Tile in the UUI.

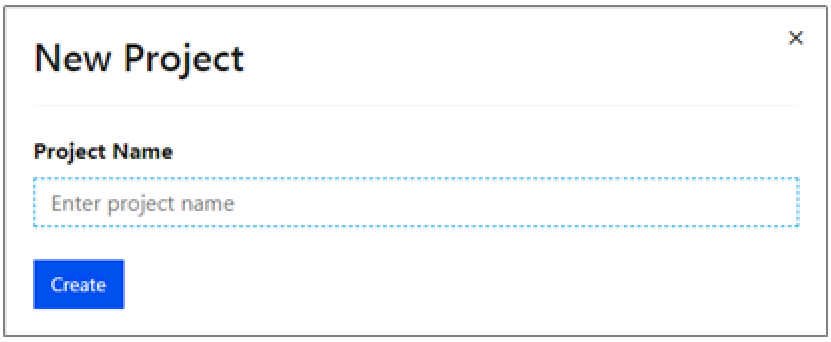

Figure 22 - New Project Dialog.

Upon selecting a project, the name of current active project is shown on the top right corner of the UUI. Users can choose not to use the project management interface at all, which results in the files to be stored in the general space for the user without any file-project relationship.

2.9.2. Integrated Tools

Since the FPM requires to override the normal behaviour of the UUI tools with regards to file storage (especially cloud storage), the integration of the FPM and UUI tools is only possible when the tool is developed by eCraft2Learn team or when the tool is open source. This means that Tinkercad, Trello and Thingiverse are not integrated with FPM system. In the following sections UUI tools which are integrated with FPM are presented and their use of FPM is described.

2.9.3. eCraft Plan

eCraft Plan allows the users to save/load their work using the save icon on the top left corner. This presents a save/load dialog that provides two different options to the user: using the cloud or local storage. The cloud option queries FPM for a list of user files for the eCraft Plan tool and their relation to the projects (Figure 23). This information is then presented in the dialog as two separate sub-lists:

- files related to current project,
- all user files

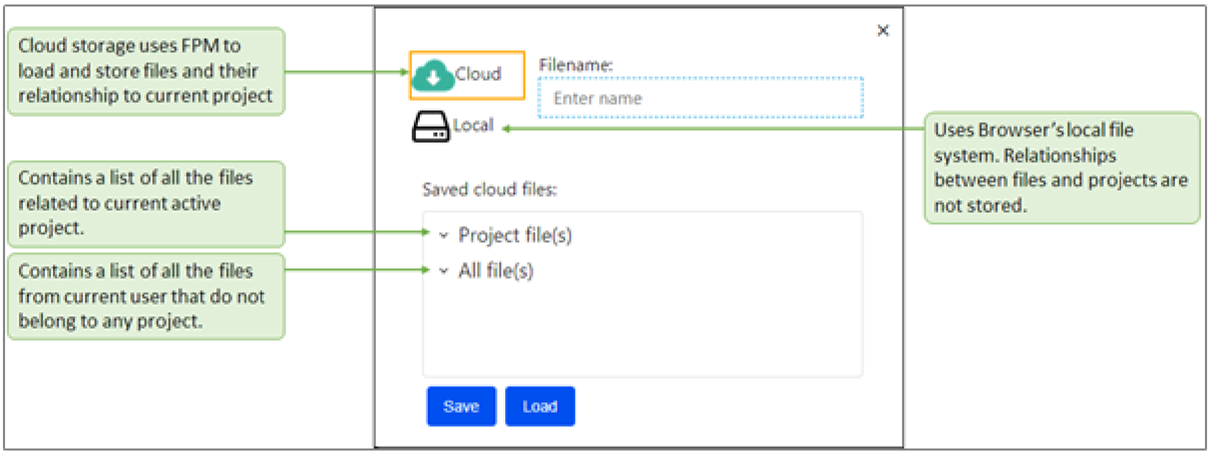

Figure 23 - eCraft Plan's save/load dialog and its interaction with the FPM.

2.9.4. eCraft TODO

eCraft TODO takes a different approach to saving files than eCraft Plan. This is because for a given project there can only be one TODO list. Therefore, the tool saves the current list whenever the “save” button is pressed and loads it automatically when a project is opened through FPM.

When a project is selected, FPM sends a list of user files and their associated tools back to the UUI. The information is then stored in the browser for further use. When the user launches the TODO tool, the tool fetches the information from the browser’s storage and if a TODO file for that project exists, TODO tools queries the FPM for the contents of the file and automatically loads it. Therefore when the tool window is opened, the user will have the TODO list for that project open and ready.

2.9.5. Snap! and Snap4Arduino

There are two ways to run Snap! and Snap4Arduino: 1) using the public servers and 2) using a local copy on a web-server. eCraft2Learn uses the second option and includes modified versions of Snap! and Snap4Arduino on eCraft2Learn’s web-servers. This solution allows us to:

- Modify Snap! and Snap4Arduino source code. This is especially useful for data analytics and FPM services in eCraft2Learn.

- In case of bad or no internet connection, run the eCraft2Learn from a local server which includes Snap! and Snap4Arduino. Please notice that in such cases (no internet) some tools will be unavailable (e.g. Tinkercad). This will not be the case with Snap! and Snap4Arduino if they are not running from a public server.

Snap! and Snap4Arduino have their own built-in cloud storage services. These services are unavailable if the tools are running on other servers than their original public servers. This means that the code should be modified to use FPM instead. Therefore Snap! and Snap4Arduino's code has been modified to communicate with FPM without any changes in the user interface. This means that the look and usage of cloud storage services of these tools stays the same but it communicates with FPM instead. The cloud option in the save/load dialog of Snap! (Figure 24) is modified to communicate with FPM to receive files which are related to current project. Please also notice the "show all files" checkbox to the top left. This is eCraft2Learn's addition to the interface which requests FPM to send all the user files back (and not only the ones which are related to this project).

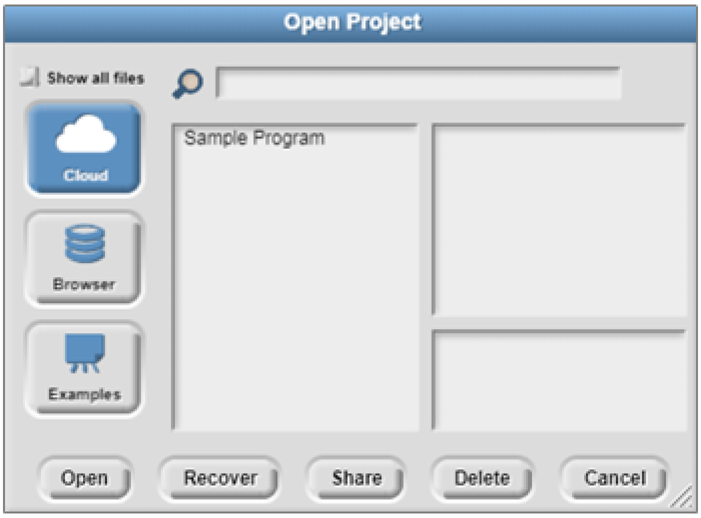

Figure 24 - Save/Load dialog in Snap!

Sharing Tool (Share My Work)

eCraft Sharing tool also uses FPM to access the user files and share them with other users of the UUI. Since the sharing tool should allow the user to share any of their files and not only the files which are related to current open project, the sharing tool always queries the FPM for all of the files related to a given user (this even includes the hidden eCraft TODO file, as the learners might want to share it with other groups or members). The user can then choose which file they want to share.

In the "Share My Work" dialog of the Sharing tool (Figure 25), it can be noted that if the user decides to share a cloud storage file (and not a local file), then the dropdown list (at the bottom of the dialog) is populated with a list of all the files retrieved from FPM. We will discuss the sharing tool in more detail in Section 2.9.

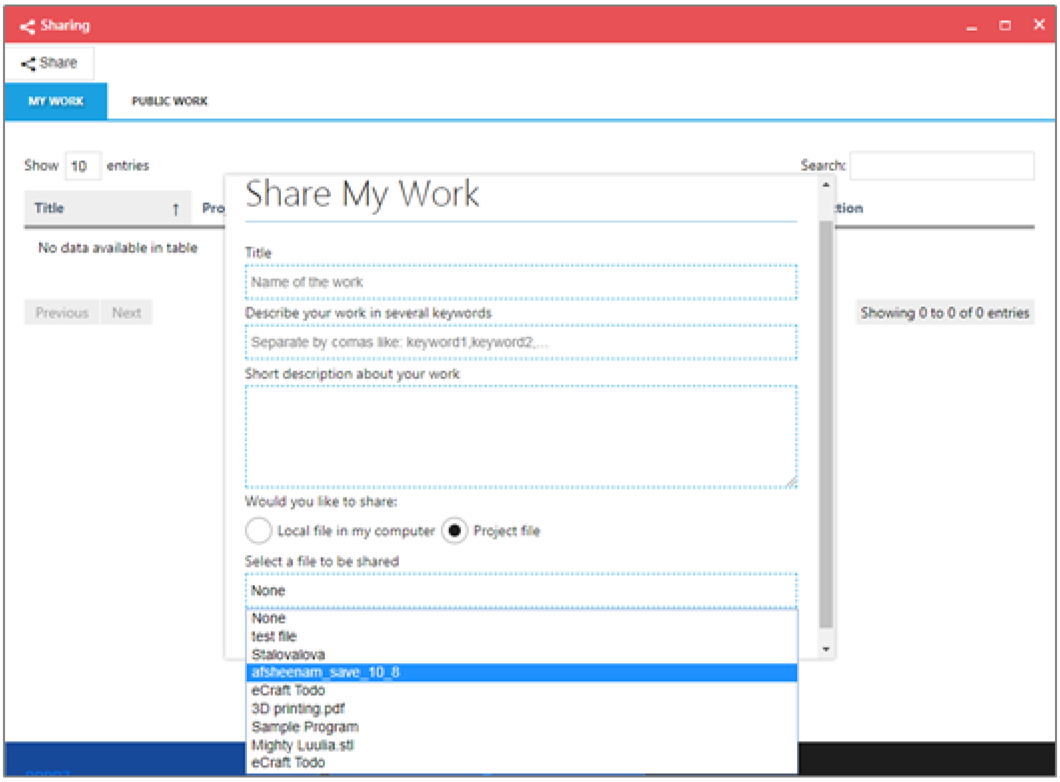

Figure 25 - eCraft2Learn's Sharing Tool.

2.10. SHARING MODULE

This module aims at allowing the users/learners to share their work created with the various tools available in the UUI. The sharing feature lets the user/learner share the work in two primary ways: a) upload it manually or b) by sharing a file that is already associated with a project in the UUI. The sharing of the work needs to be authorized by a teacher, responsible for supervising the work being carried out, to be viewed publicly within the pilot site.

The two main goals of the sharing feature are, firstly, to provide the user with the opportunity to share their work with others, secondly, to give the user a manageable overview over the work they have shared as well as an overview over the work that the users' peers have shared.

The overview of the sharing feature shows a simple system where the front end sends requests or data to a server API which in turn queries a database to retrieve the requested information (Figure 26). This information is then returned to the front-end and displayed to the user.

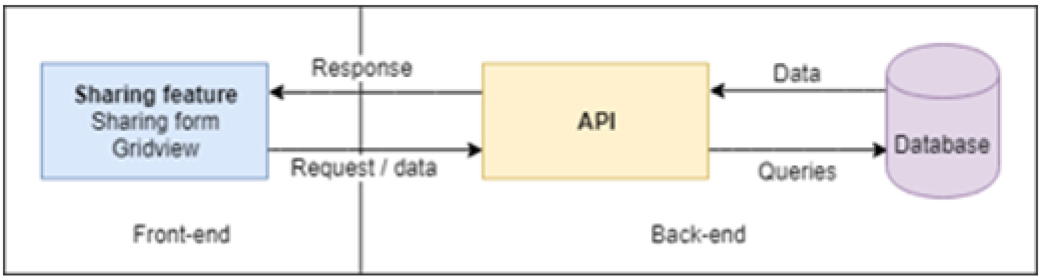

Figure 26 - A general overview of the sharing feature, with front-end and back-end.

The learner's UI for the sharing tool can be divided into two parts: 1) the sharing form (Figure 27) and 2) the shared files list (). The sharing form makes requests to the back-end to retrieve the following in the above described manner:

- A list of the tools associated with the UUI.
- A list of the users' own projects. This is handled through FPM.

The sharing form allows the students to share either a local file (Figure 27) or a project file (Figure 29) which is already stored through FPM. If they choose to share a local file, the tool and the project that the file is related to should be set manually, as this information is only available for the files which are handled through FPM. After sharing a file, the request will need a teacher's approval to make it visible to the rest of eCraft2Learn community (Figure 28).

Further the sharing form initiates sending of information to the back-end for storage in the database. Storing the file is handled by FPM while the rest of the sharing information is handled by Sharing Module's API.

The sharing tool also provides a list of shared files (by the user and/or by others) which allows the user to handle their own shared files (e.g. when they want to remove a file from being shared) or files shared by others (e.g. when a user want to download a file shared by others). When viewing and managing one's own shared files, a separate field in the list provides the user with information regarding the status of the share request (e.g. approved by teacher) (Figure 28).

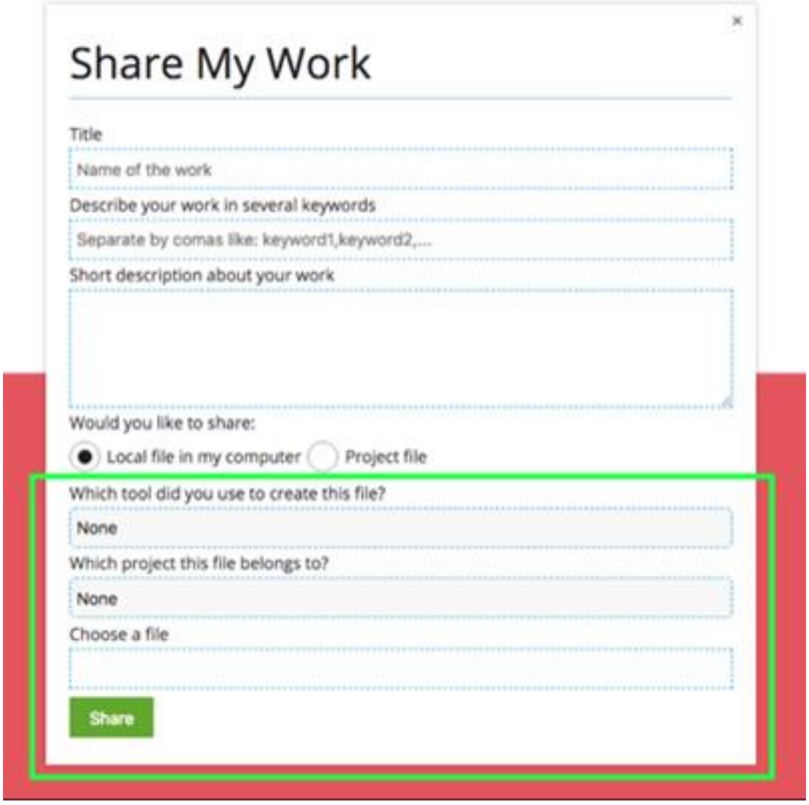

Figure 27 - The sharing form showing the local upload specific fields in green.

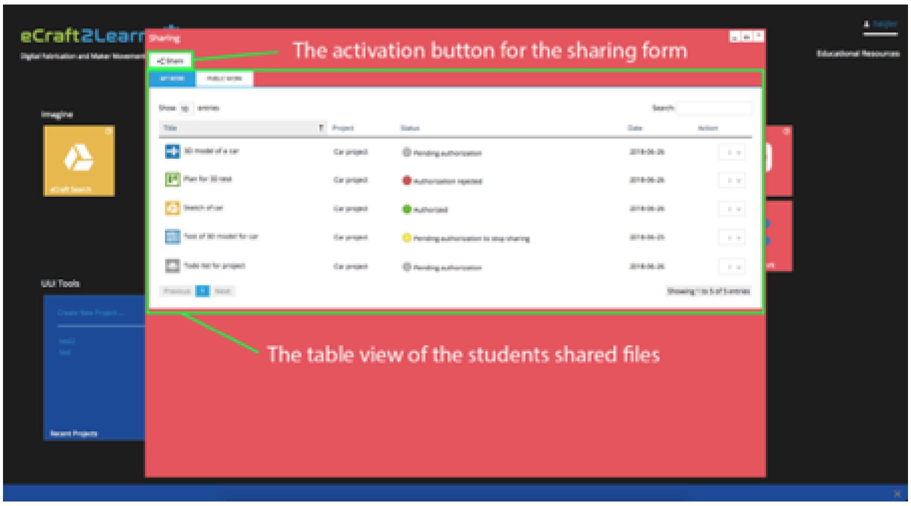

Figure 28 - The sharing interface with the shared file list open. Notice that some files are pending for approval by the teacher (yellow status) and some are rejected for sharing by the teacher (red status).

2.11. Commenting Module

The commenting module is included in Sharing module and is designed for enabling students to provide some feedback on peers' work as well as their own - with this module student can give feedback and do peer review. This is an important aspect of sharing as identified during the piloting activities. The module was designed in collaboration with teachers. For example, the five-point rating scale feedback widget to be used as comments was suggested by teachers:

⭑ ⭑ ⭑ ⭑ ⭑ - This looks smashing, awesome job

⭑ ⭑ ⭑ ⭑ - Looking really good, excellent job

⭑ ⭑ ⭑ - Way to go! Good job

⭑ ⭑ - Nice job, keep on trying for more

⭑ - Nice try, ok

The Commenting module can be found under the action menu in the Sharing tool.

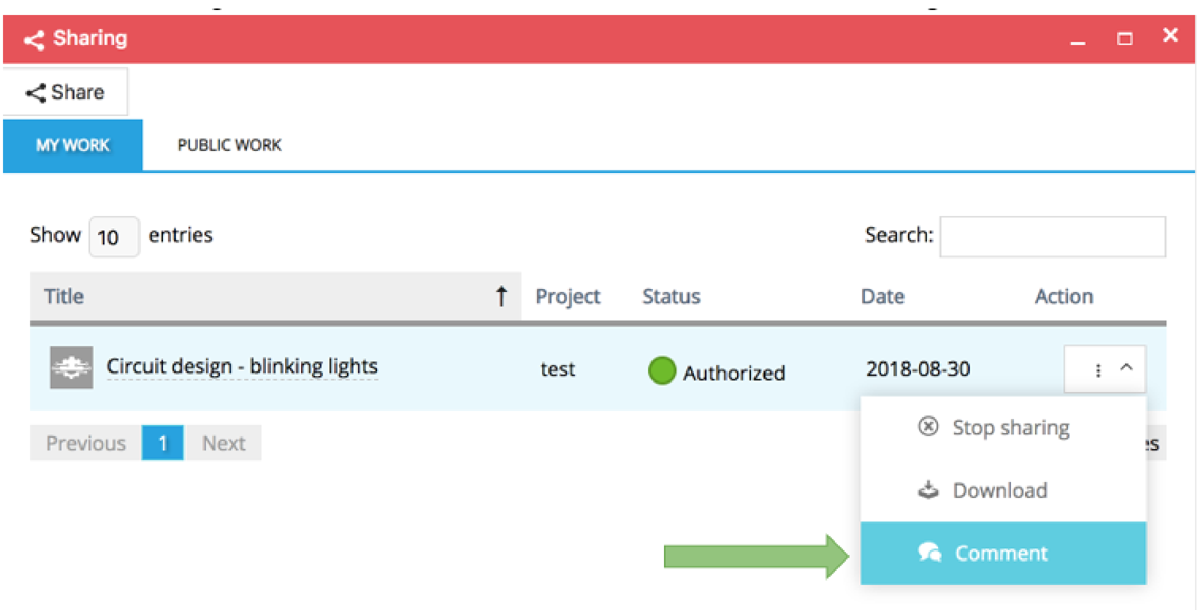

Figure 29. Access to the Commenting module (Marked with green arrow)

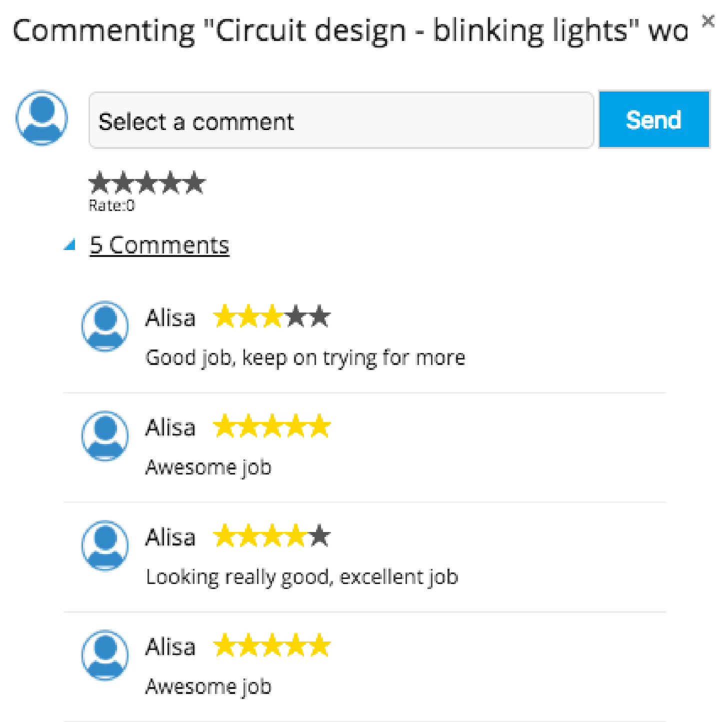

Figure 30. Commenting module main user interface

The user can select the comment from the list and press the button "Send" to send the comment as shown on Figure 31. The rating will be automatically calculated based on the selected values in the dropdown list.

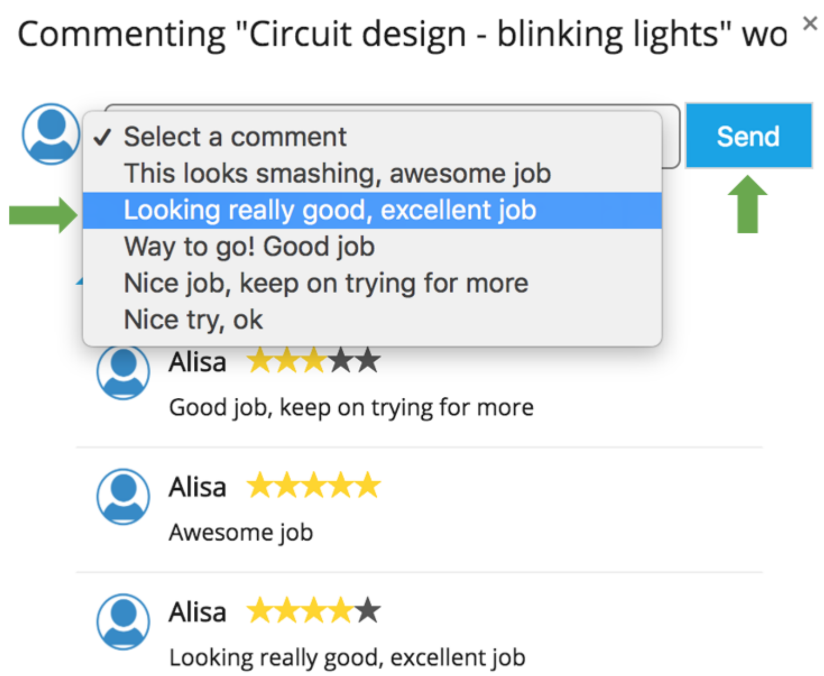

Figure 31. Selecting and sending the comment (Marked with green arrow)

The number of comments and comments that were published before are visualized as a list of comments as shown on Figure 32.

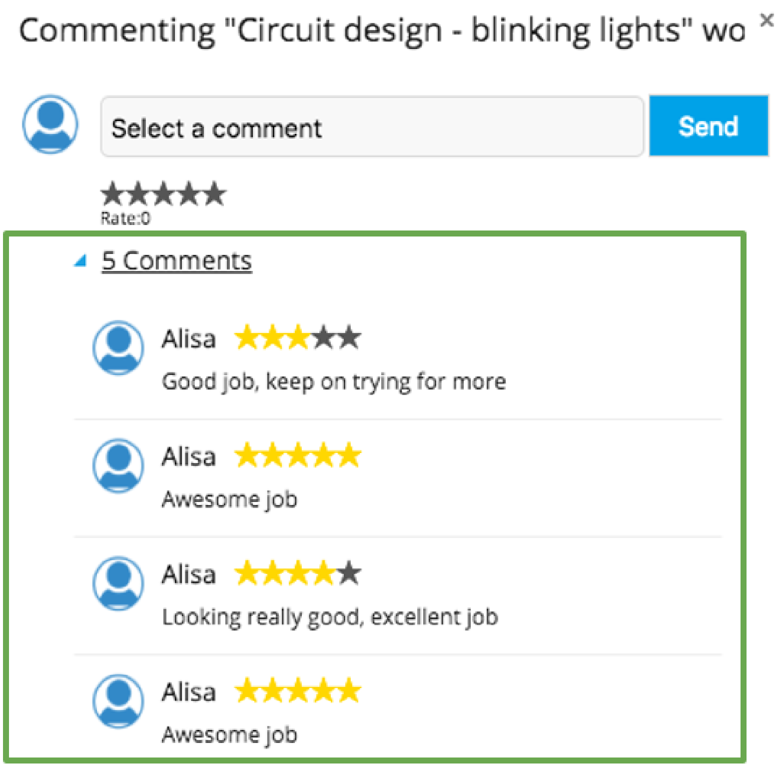

Figure 32. List of comments previously published

3 THE INTERFACE FOR TEACHERS

The teacher interface has menu items for the learning analytics, customising the UUI, adding additional educational resources, confirming the students' self-evaluation, the call-4-action module, and approving students' sharing of files. Each of these modules are described below. The teacher should initiate the interface to an already created session ID or a pilot site in order to access the database that holds the data stream from the UUI. If the teacher does not have a session ID to log in to, he/she needs to create one and give this number to the students for them to log in to that particular session ID in order for the system to be able to collect the appropriate data for the learning analytics. Many services of the teacher interface are password protected. Teachers need to register their email address and choose a password in order to prevent students and others from improper access.

3.1. Customisation of the UUI by teachers

The teacher interface has a panel where a teacher can decide which subset of the eCraft2Learn tools are available to students in a session. This is in order to support the role of the teacher as a coach and facilitator of the learning experience. By clicking on switches the teachers can toggle whether a tool is available or not (Figure 33).

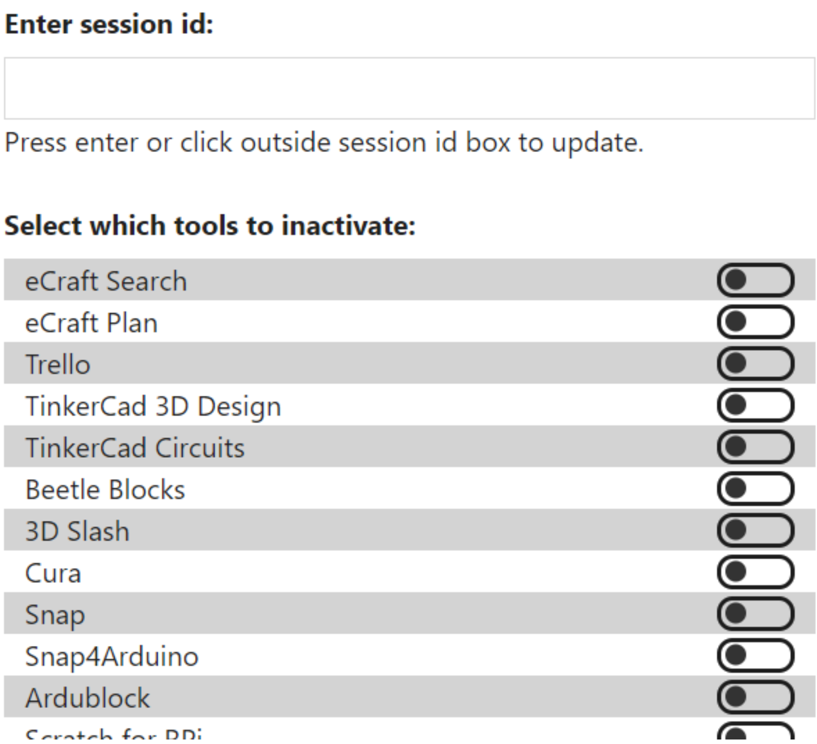

Figure 33. Part of the interface for teachers to customise the UUI

3.2. Learning Analytics

eCraft2Learn learning analytics (LA) aims at analysing the digital traces that the Unified User Interface (UUI) and some of its tools collect from the students during the sessions (Figures 34 and 35. Ultimately, the ambition is to allow the teacher to use the LA to deepen the understanding in the context where the projects take place either in formal or informal settings. LA is organized as a collection of several data mining tools found in a framework including (1) classification, (2) cluster analysis, (3) outlier detection, and finally (4) association rule learning. These algorithms are widely used in other similar tools of learning analytics and educational data mining. All data mining algorithms used in LA have been implemented so that no third party APIs have been used. Thus, the solution can be used by a school, or other education provider, without considering the limitations due to third party API solutions. For the classification and prediction, LA uses Iterative Dichotomiser 3 decision tree, for cluster analysis, LA uses Neural N-Tree neural network, for anomaly detection, LA uses 1 class support vector machine and for association rule learning, LA uses apriori algorithm.

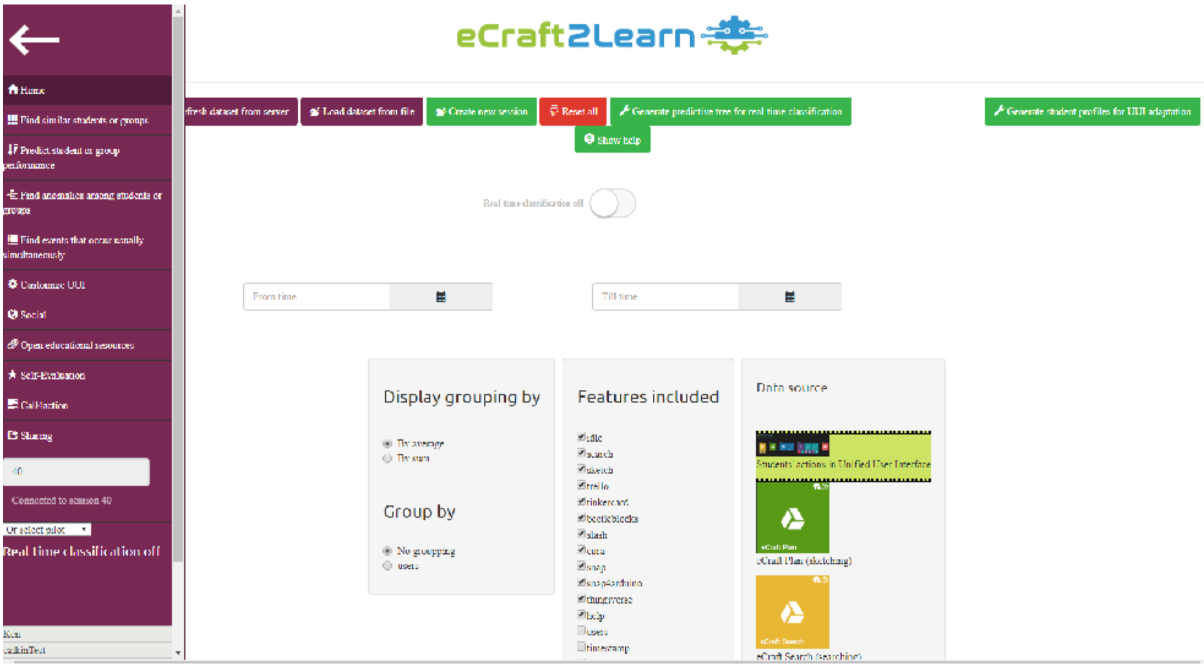

Figure 34 - Main interface of the Learning Analytics system.

Besides the above-mentioned more traditional learning analytics methods, LA introduces a white-box approach to classification and cluster analysis. White-box approach in contrast to traditional black-box approach (Neural Networks for instance) opens the data mining process by using white-box algorithms, which are easy for a human to comprehend and to interpret (decision trees, for instance). The model produced by these algorithms are visually represented to the teacher. The teacher has the ability to adjust the models, which are the built classifiers and cluster analysers and hence, the classification and cluster analysis process itself produces domain knowledge as opposed to the results of the cluster analysis and classification. Such approach aims at deepening the understanding of the learning process.

The native tools of the UUI (eCraft Search, eCraft Plan, eCraft TODO) and some third party tools such as Snap and Snap4Arduino collect the traces from the students. The traces consist of the actions that the students take inside of the UUI and its tools and passes the actions to the eCraft API in JSON format. From the eCraft API, the data vectors will be inserted into the database whereas LA makes requests to the API to get the inserted data vectors. The structure of JSON is generic and none of the features collected is hard coded to the system. Thus, in the future, the possible traces collected from the student actions can be modified according to the needs of the teachers and the other users of LA.

The analysis is based on the tools and thus, actions in every tool will be analysed separately. This is due to the dimensions of the data vectors. If every tool would be combined, then the final vectors would have over 100 dimensions, where for instance standard clustering algorithms would fail (Euclidean distance, Manhattan distance, etc.).

When the user enters the session id in the main view of LA, the dataset related to that session will be loaded automatically by LA. The dataset consists of individual actions within the UUI. For each data vector, there exists an action, timestamp and the user, who took the action. From LA, it is possible to change the data source to any of the native tools in the UUI and Snap4Arduino / Snap. The dataset can be pre-processed by grouping the dataset by a student or a group of students. Through the grouping, the dataset is more meaningful to explore, because now each vector contains the sum or the average of individual action. For instance, a certain student could have launched Snap 3 times, Search tool 2 times and so on.

After the pre-processing of the dataset, the analysis tools can be used to analyse the pre-processed dataset. If a teacher wishes to predict the performance of the students, the teacher labels some of the data vectors from the main view. After the labelling, the classification algorithm (Iterative Dichotomiser 3) renders the decision tree that corresponds the predictive model generated by the algorithm (Figure 34).

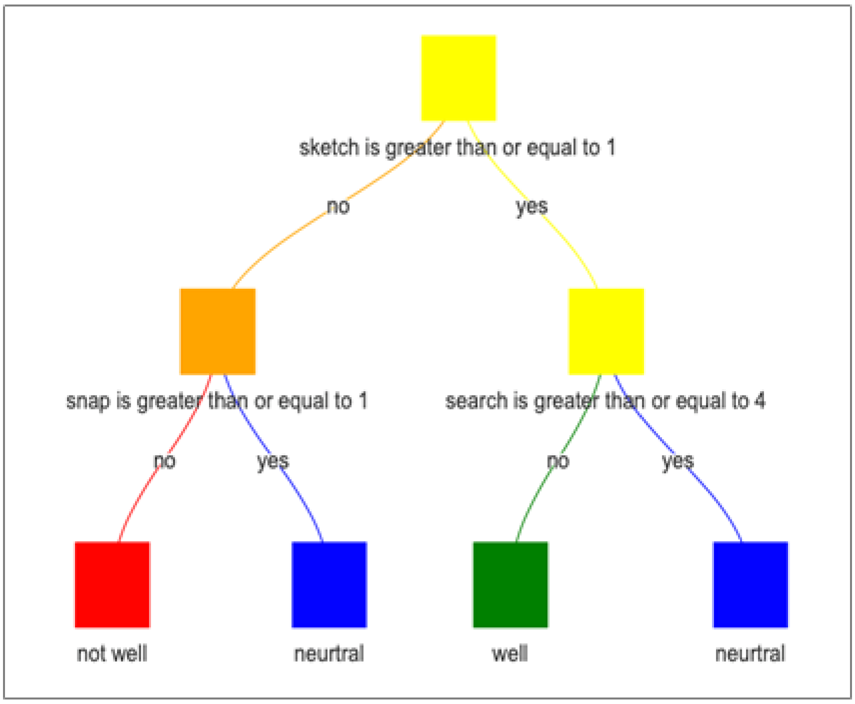

Figure 35 - Iterative Dichotomiser 3 decision tree model generated by teacher in LA.

If a teacher wishes to find anomalies among the students, the teacher has to label the students from the main view as “Anomaly”, “Not an anomaly” or “Not included”. After the labelling, support vector machine processes the dataset and finds the parameters that best fit the dataset based on the labels. Afterwards, a table view is rendered to the teacher, where each row represent a student marked either red or green where red means that the student is an anomaly. That is, the behaviour of that student greatly differs from the rest of the dataset (Figure 36).

0	0	0	0	0	0	0	0	0	1	1	kostasasimakopoulos@gmail.com	Mon Apr 09 2018 01:07:02 GMT+0300 (EEST)	0
4	47	2	3	1	0	0	1	37	5	18	Testaaja	Mon Apr 09 2018 05:30:58 GMT+0300 (EEST)	0
0	0	0	0	0	0	0	0	3	0	2	Ilkka Roininen	Mon Apr 09 2018 11:10:14 GMT+0300 (EEST)	0
0	0	0	0	0	0	0	1	0	0	1	Testaaja	Mon Apr 09 2018 11:10:29 GMT+0300 (EEST)	0
0	3	1	4	1	0	0	4	20	1	5	arstronautit	Tue Apr 10 2018 10:11:57 GMT+0300 (EEST)	0

Figure 36 - Anomaly detection table.

The remaining two analysis methods are cluster analysis algorithm that groups the similarly behaving students together and association rule mining algorithm apriori that draws a directed graph from the actions taken by all students usually occurring simultaneously (Figure 37).

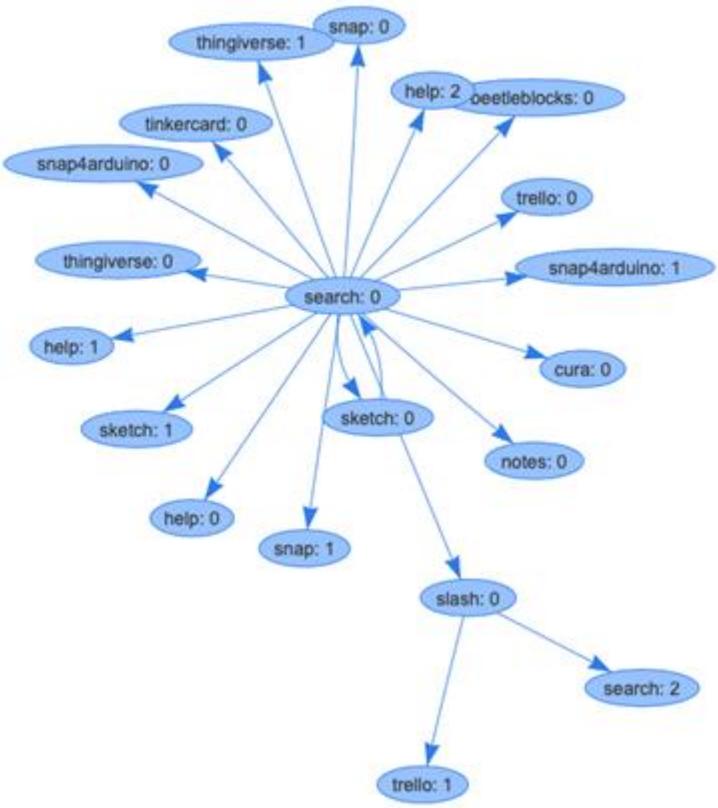

Figure 37 - Apriori graph from students' actions displaying number of uses of each tool.

3.3. CALL-4-ACTION MODULE

This module is designed for teachers and it allows requesting students to do certain activities within their own sites while exploiting the tools available within the UUI. With help of this module, teachers will be able to create a specific call for action and assign it to students. Accordingly, these actions will appear in student's UUI ensuring that the teachers are able to communicate the requests to their students. Examples of actions: “Reflect about your work”, “Share your work”, “Use tool eCraft Plan to create the plan of your work”, etc.

There is a scenario when one task can depend on another task. In this case, two types of tasks are suggested:

- Independent Task - is a task that can be completed and does not require other tasks to be done.
- Dependent Task - is a task that depends on the previous ones and will be available only after prerequisite were completed. There is also a Reflection Task that will open separate window in UUI where students can reflect on their work.

Each task has two states:

- Pending - for a task initiated by the teacher.
- Completed - for the phase in which the student declares that his/her work on a task is completed.

This module was implemented for both teachers to create tasks and students to view the corresponding tasks. It is deployed on the eCraft2Learn learning analytics git repository. Below the general architecture of the Call4Action module consists of two communicating parts: the student's UUI and the teacher's interface.

The teacher user interface (Figure 38) allows the teacher to create a new task and assign it to all or specific students. This can be a general task (e.g. share your designs) or a reflection task. Reflection tasks are specific tasks which require the students to reflect on their learnings and progress. This type of task will activate a special user interface on the student side which is discussed in the next section.

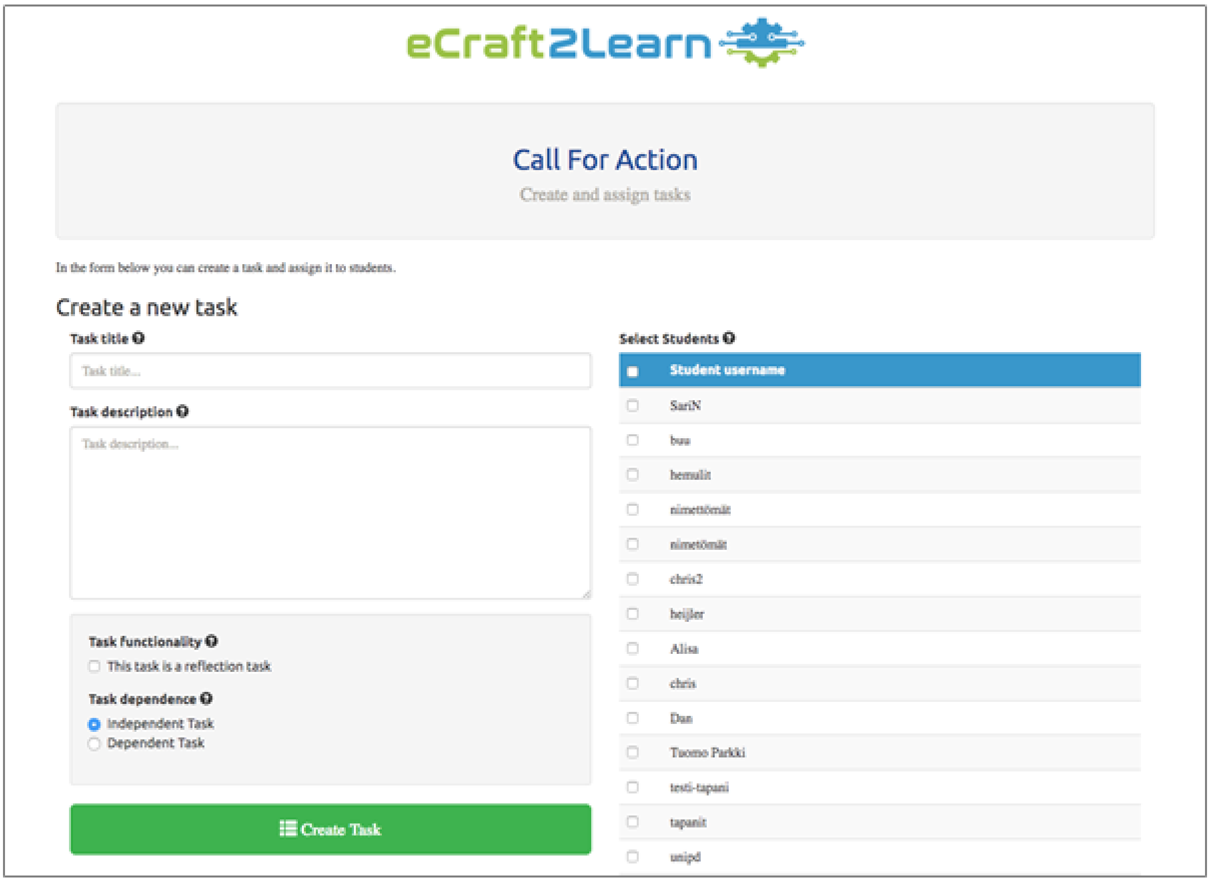

Figure 38 - Teacher's user interface for Call4Action module.

The actions that teachers create and assign to students will appear on the student's UUI in "Tasks List" panel (Figure 39). After the students login to the UUI, the system will load all actions (completed, not completed) that are assigned to the student. Every action in the list is expandable, the first action is expanded by default. Each action has a name (title), short description and a "Complete Action" button.

After a student completes an action he/she should press button "Complete Action" to notify the teacher that the task is performed. In case the student presses the button "Complete Action" by mistake, then he/she can undo this operation by clicking the "Undo" button (Figure 39).

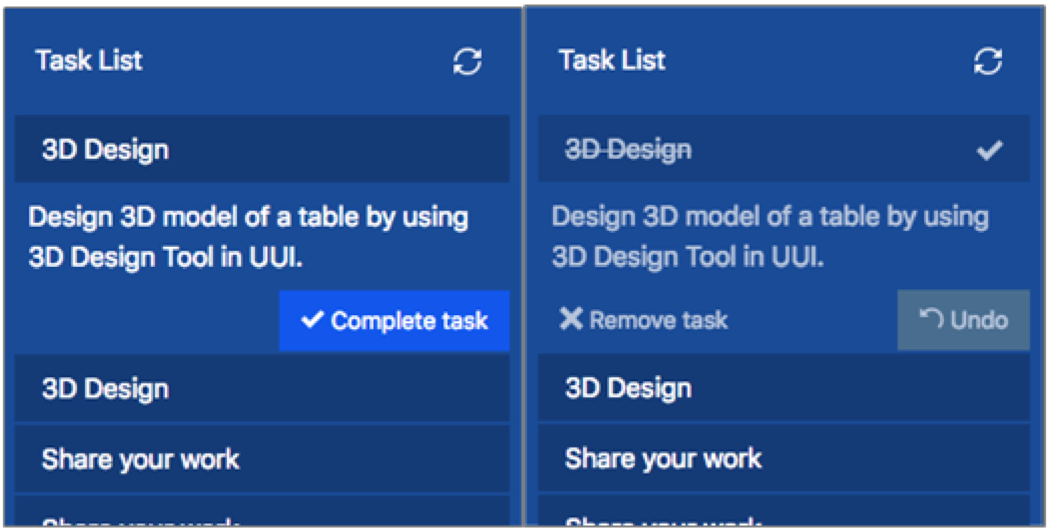

Figure 39 – Task list tile in the UUI.

The "Remove task" button appears only for completed tasks and it means that the task will not be visible for the student in the Task List. Dependent Tasks has additional text in the description of the task notifying the students that there is another task which should be performed first. Also the "Complete task" button is disabled for dependent tasks as long as the dependencies are not resolved.

As mentioned before, a specific type of task is the reflection task. These tasks have an auto generated link that opens the Reflection window on the bottom right corner (Figure 40).

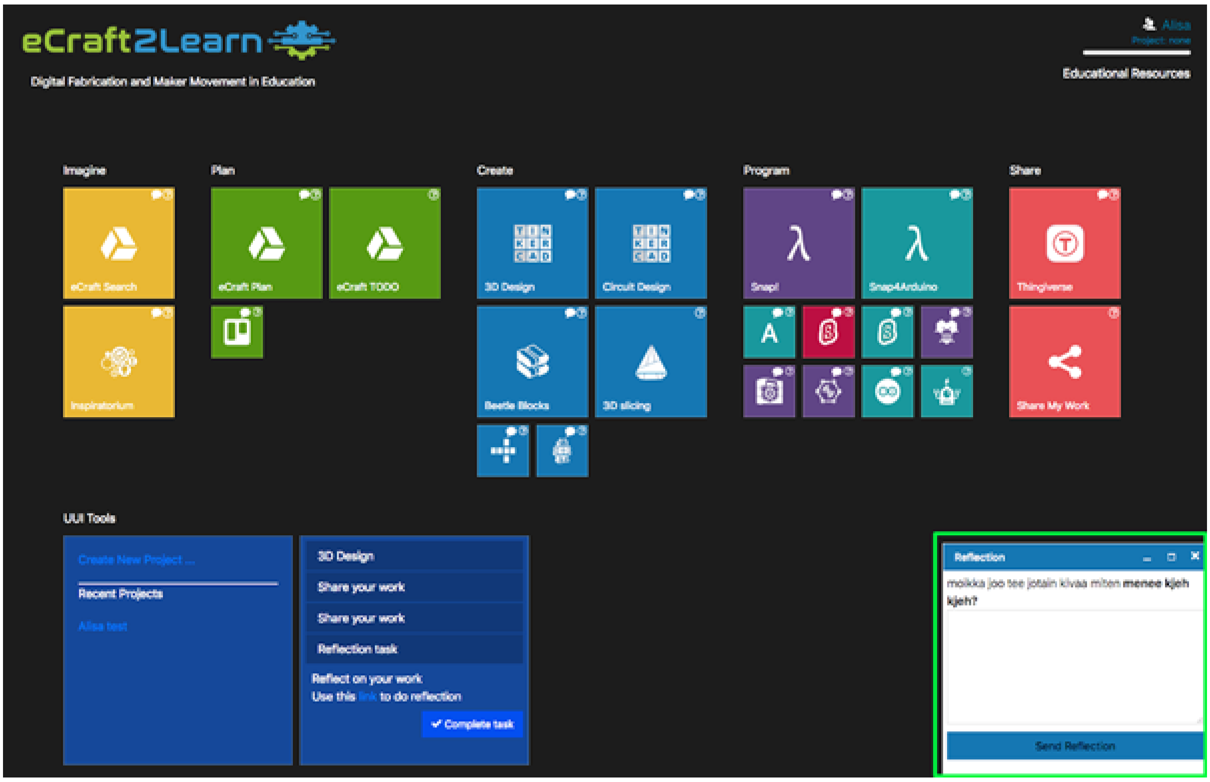

Figure 40 - Reflection window activated through reflection task from action panel (marked in green).

4. Conclusion

This document and accompanying videos demonstrated the third version of the unified user interface. This version demonstrates integration with the educational extension, thus the tools to support the five stages of eCraft2Learn methodology. The set of functionalities were selected to provide support for the rounds of pilot tests. The results of the pilot studies continued to provide useful insight into further development of the UUI as well as the rest of the system.

5. References

Literature:

Barata, G., et al. (2013). Improving participation and learning with gamification. Proceedings of the First International Conference on Gameful Design, Research, and Applications. Toronto, Ontario, Canada, ACM: 10-17.

Deterding, S., et al. (2011). From game design elements to gamefulness: defining "gamification". Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments. Tampere, Finland, ACM: 9-15.

Dicheva, D., et al. (2015). Gamification in Education: A Systematic Mapping Study.

Dominguez, A., et al. (2013). "Gamifying Learning Experiences: Practical Implications and Outcomes." Computers & Education **63**: 380-392.

Gibson, D., et al. (2015). "Digital badges in education." Education and Information Technologies **20**(2): 403-410.

Hakulinen, L. and T. Auvinen (2014). The Effect of Gamification on Students with Different Achievement Goal Orientations.

McLoughlin, C. and M. J. W. Lee (2010). "Personalised and self regulated learning in the Web 2.0 era: International exemplars of innovative pedagogy using social software." Australasian Journal of Educational Technology **26**(1).

Miliband, D. (2006). Choice and Voice in Personalised Learning, OEDC/CERI.

Peirce, N., et al. (2008). Adaptive Educational Games: Providing Non-invasive Personalised Learning Experiences.

Pirker, J., et al. (2014). Motivational active learning: engaging university students in computer science education. Proceedings of the 2014 conference on Innovation & technology in computer science education. Uppsala, Sweden, ACM: 297-302.

Voigt, C., et al. (2018). Hacking the Knowledge of Maker Communities in Support of 21st Century Education.

Websites:

https://codemirror.net/, retrieved on 29.10.2018

https://mosquitto.org/, retrieved on 29.10.2018

https://github.com/ecraft2learn/craftbot-for-arduino, retrieved on 29.10.2018