A Gamified Platform for Requirements Engineering Education

Incorporating narrative elements and pedagogical principles

Lappeenranta–Lahti University of Technology LUT

Master's Program in Software Engineering & Digital Transformation

2022

Carlos Julia

Examiner(s): Assistant Professor Antti Knutas D.Sc.

Associate Professor Dominik Siemon

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

LUT School of Engineering Science

Software Engineering

Carlos Armando Julia Fandiño

A Gamified Platform for Requirements Engineering Education

Master's thesis

2022

56 pages, 6 figures, 6 tables and 5 appendices

Examiners: Assistant Professor Antti Knutas D.Sc. and Associate Professor Dominik Siemon

Keywords: Requirements Engineering, Learning Management Systems, Gamification, Pedagogy, EdTech, Design Research

Requirements engineering aims at preventing the failure of software development projects by building the right product. In recent years the training of knowledge and skills of this field has been at the center of research. Recent trends in requirements engineering education have focused on tools and techniques that revolve around role playing, problem solving and simulation, however, no approach has incorporated narrative theory or solid pedagogical principles. This work presents a solution that incorporates pedagogical practices and includes gamified design patterns based on narrative structures found in storytelling and plots. To do so, it approaches the problem with design science research methodology, creating an artifact and subjecting it to user testing. The result is a learning system that displays the features intended to fill a vacuum existing in the field of requirements engineering education.

ACKNOWLEDGEMENTS

To my dear wife Masha and to my family, their support has always been priceless.

SYMBOLS AND ABBREVIATIONS

Abbreviations

API	Application Programming Interface
FURPS	Functionality, Usability, Reliability, Performance & Supportability
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IEEE	Institute of Electrical and Electronics Engineers
INVEST	Independent, Negotiable, Valuable, Estimable, Small and Testable
ISO	International Standards Organization
LMS	Learning Management System
RE	Requirements Engineering
REE	Requirements Engineering Education
REST	Representational State Transfer
SaaS	Software as a Service
SWEBOK	Software Engineering Body of Knowledge

Table of contents

Figures

Tables

1 Introduction

Building the wrong solution can lead to the failure of software projects. Some first and second order consequences of these situation can be overhead expenses, release date delays, difficulties in maintainability, lost market opportunities, team frustration and internal divisions among other financial, commercial, and human ramifications in organizations. Requirements engineering exists to address and prevent these scenarios. However, weak qualifications and lack of experience have been identified as some of the causes for hidden or incomplete requirements when performing the processes of RE in companies (Fernández *et al.*, 2017). In this sense, requirements engineering education has a tangible impact on the performance of organizations with projects that involve software development.

Although RE is a common subject across universities that offer software engineering education, some students tend to see it as an unimportant or unattractive topic (García *et al.*, 2020). This may lead organizations that hire recent graduates to face project management issues associated with the topic, for example building the wrong solution, not developing necessary software features, creating a UI/UX that is not user friendly, etc. Such issues could be better prevented with training. On the other hand, gamification has been explored by past research to improve student engagement, but little pedagogical background has been observed in the design or structure of gamified solutions aimed at teaching RE. Moreover, roleplaying as a game design element has been identified in REE systems and is not uncommon in classical RE education, however little attention has been paid to the narrative aspects of roleplaying. Storytelling elements might hold the potential for further engagement and could improve the perception of the role of requirements engineering among learners.

1.1 Thesis Goal & Scope

The objective (O) of this work is to design a gamified system of requirements engineering skills training. The research questions intended to be answered are:

- RQ1: Can roleplaying and/or storytelling be implemented in an online learning system?
- RQ2: What key skills would be necessary for comprehensive training of requirements engineering for engineering students?
- RQ3: How could such a system facilitate the teaching of key skills?

The intended approach is to review relevant literature exploring relevant concepts and applying design science to elaborate an artifact which will be tested with degree software engineering students.

The artifact will be created using the Django framework for python 3.0 programming language, however, the artifact will be aimed at testing features, for this reason the user interface will not be comprehensibly developed from the aesthetical point of view. Additionally, since the artifact will work as a proof of concept, the content elaborated will not cover the totality of teachable subjects that fall under REE, it will only have enough information to evaluate the objective and research questions set and thus it will neither be a serious game nor a full-fledged e-learning gamified course although the result might resemble the later.

1.2 Structure

Section 2 will explore the related literature, in particular the concepts of requirements engineering, its standards, and bodies of knowledge as well as the research associated with the teaching of RE. A pedagogical background will be displayed based on the concepts of mastery learning and Merrill's' principles of learning. Basic concepts of gamification and storytelling will be introduced, and a current existing solution will be examined.

Section 3 will describe the methodology and reasoning involved in the design choices set for the artifact as well as its architecture, requirements, procedures for elaboration and measurement strategies. Section 4 will present the evaluation process, its findings and the analysis of the findings. Section 5 will discuss the results, explore the implications and contrast them with an existing solution that shares a similar objective. Section 6 will present the conclusions reached in this research.

2 Related Works

This section presents a literature review of academic works pertaining to the core subject of this research. The keywords selected were intended to cover 3 main aspects: literature related to requirements engineering, pedagogical practices, and gamification. To gather the related works, the following terms and sub combinations of them were sought in academic databases:

- requirements engineering (cost OR expense)
- requirements engineering learning
- requirements engineering education
- requirements engineering teaching
- mastery learning
- principles of learning
- storytelling gamification education
- gamification framework
- gamif*[1]
- systematic mapping
- state of the art
- literature review

The main search engines utilized for this purpose were LUT Primo and Google Scholar. Additional queries and keywords that returned no valuable resources have not been included in the list above. The sources include articles and books listed in Ebsco, IEEE, Elsevier & Springer. A total of 36 sources were reviewed of which several have been directly referred

[1] reflecting gamification, gamified and similar terms

to in this work while others are not referenced directly but constitute part of the theoretical base laid here.

2.1 Requirements Engineering (RE)

Requirements engineering (RE) is the practice of identifying the needs of organizations to establish what tasks a system must perform at the end of a software development project. The essential steps of the process are comprised of “elicitation, analysis, specification, and validation of software requirements” (Bourque, Fairley, and IEEE Computer Society, 2014).

Requirements’ elicitation consists of applying techniques to extract relevant information from carefully selected stakeholders to determine what are the requirements necessary for a given software development project. Requirements’ analysis relates to the classification, modeling, and architectural design of the requirements. Requirements’ specification is a phase when the system and functional requirements are laid down in a structured document for its later interpretation. During requirements validation, engineers establish who and how must review the collected requirements to understand whether the requirements gathered will satisfy the needs of the stakeholders. To that end, prototypes can be built, and acceptance tests should be agreed upon (Bourque, Fairley, and IEEE Computer Society, 2014).

RE is essential for software project success. Developing software is a long and costly process. Developing the wrong software may result in overhead expenses due to necessary corrections down the road. Another consequence might be to lose opportunities for failure to be “first to market” when offering a new product since organizations may cede market share to competition by missing launch deadlines. Failure to launch in time plus overhead expenses may lead to the cancellation of projects, resulting in lost money, lost effort, frustration and even friction between product management and development teams (Sommerville, 2016).

Skills and knowledge of engineers are linked to requirements engineering accuracy and success. Steps such as requirements elicitation involve activities like focus interviews, and field observation. These activities in turn demand; social abilities to interact with stakeholders, the capacity to create structured forms and the redaction of bias-free questions among other skills. Lacking precise communication may lead to misinterpretation which can surface later as unrequested or missing features in the system (Sommerville, 2016).

2.1.1 Bodies of knowledge & standards

Several curricula, bodies of knowledge and standards have been identified as guidelines for the individual items that can be taught in requirements engineering education (REE), these include the Joint IEEE/ACM Computing Curricula for software engineering and computer science (CCSE, CCCS), the ACM/AIS Curriculum Guidelines for Undergraduate Degree Programs in Information Systems (IS), the Software Engineering Body of Knowledge (SWEBOK), the Software Engineering Education Knowledge (SEEK), the Requirements Engineering Body of Knowledge (REBOK), as well as IEEE standards 1233-98, 1465-98, 830-98 and ISO/IEC standards 25030:2007, 15288 and TR 24766:2009 (Ouhbi et al., 2015).

Topic	Subtopic
1. *Software Requirements Fundamentals*	1.1. Definition of a Software Requirement 1.2. Product and Process Requirements 1.3. Functional and Nonfunctional Requirements 1.4. Emergent Properties 1.5. Quantifiable Requirements 1.6. System Requirements and Software Requirements
2. *Requirements Process*	2.1. Process Models 2.2. Process Actors 2.3. Process Support and Management

	2.4. Process Quality and Improvement
3. *Requirements Elicitation*	3.1. Requirements Sources
	3.2. Elicitation Techniques
4. *Requirements Analysis*	4.1. Requirements Classification
	4.2. Conceptual Modeling
	4.3. Architectural Design and Requirements Allocation
	4.4. Requirements Negotiation
	4.5. Formal Analysis
5. *Requirements Specification*	5.1. System Definition Document
	5.2. System Requirements Specification
	5.3. Software Requirements Specification
6. *Requirements Validation*	6.1. Requirements Reviews
	6.2. Prototyping
	6.3. Model Validation
	6.4. Acceptance Tests
7. *Practical Considerations*	7.1. Iterative Nature of the Requirements Process
	7.2. Change Management
	7.3. Requirements Attributes
	7.4. Requirements Tracing
	7.5. Measuring Requirements
8. *Software Requirements Tools*	

Table 1 Topics of Requirements Engineering (Bourque, 2014) - modified from source

In the field of requirements engineering education, the O/IEC/IEEE 29148:2011 standard has been used to teach the fundamentals of requirement engineering by means of a serious game (García et al., 2020). This standard is intended to guide the redaction of documents and collection of knowledge in a homogeneous and detailed manner. However, a more recent version has been published in 2018 and could potentially be used as an updated and detailed guide to design educative material for the field of requirements engineering.

2.1.2 Requirements Engineering Education (REE)

Researchers have been trying to find adequate ways to teach requirements engineering in recent years. A systematic literature review found 152 academic publications in the field of requirements engineering education between 1988 and 2020. A large majority of the papers were published after 2007 and most of them were either solution proposals or experience reports (Daun, Grubb and Tenbergen, 2021). Although Strategies and approaches for REE have been proposed, Daun, Grubb and Tenbergen (2021, p. 265) concluded:

> *Our findings suggest that very few studies deal with instructing requirements consistency, traceability, safety, and security. Moreover, RE education presently suffers from a lack of a common pedagogical basis and systematically gathered evidence.*

Several researchers have approached the teaching of requirements engineering using simulations by giving tasks to students intended to recreate real life projects and challenges. This is done with the intention of helping students to develop some of the skills related to the field such as communication and teamwork as well as to allow the pupils to familiarize themselves with the basic concepts, steps and documentation that comprise the practice of requirements engineering. Efforts have been made to approach teaching in a way that closes the gap between academic views and real life needs by introducing the students to formal methods (Gibson, 2000) (Connor, Buchan and Petrova, 2009). Other approaches have shown that students involved in role-playing have demonstrated a better performance over their peers who did not engage in roleplaying (Svensson and Regnell, 2017). REE researchers have gone as far as implementing gamified platforms to teach specific subsets of knowledge within the field such as security requirements engineering with positive results (Alami and Dalpiaz, 2017).

2.2 Mastery learning & principles of learning

A pedagogical approach should be included as a design feature of any system built to teach. Mastery learning is a robust approach with a history of validation while Merrill's principles of instruction provide a framework for coursework design.

2.2.1 Mastery learning

Mastery learning is based on the notion that students should not progress to the next stage of learning until the essential skills or concepts of the current stage have been mastered. In this sense as an example, a student of requirements engineering should not learn requirements analysis until he or she has mastered requirements elicitation. This class management strategy has existed in one way or another since the early decades of the 20th century, however it was not until the decade of 1960 that researchers defined it and began measuring its effectiveness. There are two main currents of mastery learning: personalized system of instruction (PSI) and learning for mastery (LFM), the key difference is that PSI is built towards self-paced studying where the student works with written guides while in LFM the class is guided by an instructor at a homogeneous rate but assisting students that fall behind.

According to the theory, mastery learning should work in such a way that 90% of the students in a class could perform as well as the top 10% of an alternative class without mastery learning, however even though it has been proved to work specially well for students that fall behind, allowing most students to reach such a high performance might be an overstatement, nevertheless the improvement has been measured as 0,5 standard deviation by researchers (Kulik, Kulik and Bangert-Drowns, 1990). In recent times mastery learning has been validated by direct observation of its application in online learning systems (Ritter et al., 2016).

2.2.2 Principles of learning

Merrill's first principles of learning are a condensation of the similarities between several theories regarding the optimal conditions and activities that lead to learning from a point of view of coursework design.

The first principle states that to learn, students must be presented with a real-life problem to be solved. The first step must also involve demonstrating to the learner the result that will be expected (the solution) and learning is more effective if the problems presented are shown in progression that is compared in explicit fashion (Merrill, 2002).

The second principle states that it is easier to learn when previous related knowledge is recalled for the student, that is recollecting the old knowledge that may serve as foundation or building block to the prospective solution, for example, a child learning to multiply should be instigated to remember addition since multiplication is associated with addition. As part of this principle, it is also helpful to present new knowledge as well as a structure to organize it (Merrill, 2002).

The third principle consists of demonstrating a solution rather than just mentioning it or just talking about it. The demonstration must be aligned with the learning objective and can be detailed down to examples, demonstrations, visualizations, and modeling. It is important to present relevant information and use a variety of representations and demonstrations rather than just a few (Merrill, 2002).

The fourth principle is applied when the learner is led to use new-found knowledge or skill to find solutions for problems. Practice must be consistent with learning goals; feedback must be given but the educator must gradually withdraw feedback until it is not necessary anymore. Learning is more effective when the learner is exposed to a variety of problems in sequence rather than just one individual problem or a monotonous sequence devoid of novel challenges or levels of hardship (Merrill, 2002).

The fifth principle consists of leading the learner to use their new knowledge in their routine. It is helpful to allow the learner to show the skill to others. Discussing or sharing a reflection

is also a useful way to achieve this. Finding new ways to apply the knowledge or skill promotes the integration of it into the student's life (Merrill, 2002).

2.3 Gamification of e-learning systems

Gamification and games have the potential to enhance the design of information systems, helping creators attain their objectives. In REE, gamification and serious games have been used as a strategy to achieve learning outcomes. Most authors in the field of REE have so far proposed or highlighted approaches based on simulation, role playing and gamification with an emphasis on following a standard process of requirements elicitation, gathering, specification & validation (Soo and Aris, 2018). Nevertheless, very few researchers have identified or addressed problems related to the pedagogical aspects of the skills needed to perform common tasks and activities of RE.

There are several ways to define gamification but a common accepted definition in academia states that gamification entails applying game design patterns, elements, principles, models, methods, or mechanisms to non-game systems (Deterding et al., 2011). Such components may include but are not limited to:

- Clear goals
- Leaderboards
- Badges
- Roleplaying
- Narrative storylines
- Turns
- Resources
- Time limitations
- Points

- Social engagement
- Collection of items
- Achievements
- Levels
- Rewards

Additional game elements have been proposed specifically for gamification in an educational context (Toda et al., 2019).

There is no universal objective in gamification since it is a means to achieve that which the designer intends to improve for the end user (Landers et al., 2018). A benefit of gamification could be to enhance the experience of the end user, leading him/her to specific behaviors, greater enjoyment, and increased frequency of interaction. This entails adopting a theory of motivation and identifying the component factors of internal and external motivation (Seaborn and Fels, 2015).

When narrative, social components are added to gamify education processes or systems, an increase in the attainment of cognitive, behavioral, and motivational learning outcomes has been observed (Sailer and Homner, 2020). A significant positive effect of gamification has been noted on engagement indicators across several online learning platforms (Looyestyn et al., 2017) and it has also been used to effectively increase completion rates of massive online open courses (de Marcos-Ortega et al., 2020).

2.3.1 Storytelling & Roleplaying

Storytelling and roleplaying are recognized as features of gamification that could help with learning outcomes. Storytelling consists of including narrative elements in the design of an application. An example implementation of storytelling would follow any pattern recognized in theories of narrative such as the monomyth or hero's journey, a theory describing a

standard structure for the myths found across civilizations (Campbell, 2008). The strategy to implement such elements involves focusing the narrative in either of two objectives: plot building or character building. Plot building consists of creating a narrative tension by presenting a series of events in a logical order that keeps the observer engaged by exploiting a sense of anticipation and curiosity. Character building requires the presentation and development of a character meant to create a feeling of identification in the observer. A plot has 3 basic sections, first comes setup; a challenge that must be solved. In the setup the characters are introduced, and their intent presented, then comes the rising action, which is having the characters performing actions that are consistent with their intentions, however, as they do so the characters face barriers that impede characters from fulfilling their intentions. This situation should lead to a change in mindset and human relations among the characters. Finally comes the climax, a narrative phase where the logical consequences of the actions of the characters are carried out to a conclusion where the remaining mysteries or lingering questions of the plot are clarified for the audience (Tobias, 2012).

Roleplaying happens when the observer is meant to participate in the narrative by deciding the actions of a character and witnessing the impact of his decisions on the character. The use of roleplaying has frequently been used in educational systems and research engineering education. An increase in student class attendance and higher marks have been observed in a system gamified to demonstrate narrative and roleplaying characteristics (Topîrceanu, 2017).

Narrative theory in game design is a four-dimensional model that seeks to describe the essential storytelling components of games. It states that not all games have a story but when they do they have 4 common general components split across 2 poles, the elements are comprised of: Agents, events, worlds and objects. Agents refer to the conscious entities that perform behaviors and interact with the player. Events are the circumstances that lead to the perception of a story and are described in the model in terms of the degree of agency of the player over them, from fully plotted, representing games with no agency from the player (I.e.: an interactive novel) to no kernels which are pure open game with no plot (I.e.: Tetris). Worlds represent the explorable simulation of a physical environment that can be described

through geometry and has a boundary. Objects are simulated elements and are also understood in terms of how much agency a player has on them. The two poles from narrative theory are the narrative pole, the axis of the models that describe game closer to be pure story and the ludic pole which describes games closer to be pure mechanics (Aarseth, 2012).

2.4 Current Solutions

One notorious approach in the field of REE is a serious game (Requengin) designed to teach the ISO/IEC/IEEE 29148:2011 technical specification intended to guide the process of requirements engineering. The game runs a simulation of a character tasked with the responsibility of gathering requirements for a library project, analyzing them, and then submitting the respective documentation (such as use cases) for evaluation. To date, it could be considered the most comprehensive system to teach requirements engineering that incorporates game design features, and role-playing storytelling, the results presented high self-reported levels of motivation and achievement of learning outcomes (García *et al.*, 2020).

Despite the results, Requengin and the paper presenting it have a few potential shortcomings. The authors state that they based the game in a 2011 standard since their study was carried out in 2017, however they also acknowledge that since 2018 there has been an updated version of the ISO/IEC/IEEE standard governing RE. Another characteristic of the system is that it behaves partially like a mechanical turk since a professor evaluates the submissions of the students. It should be noted that this system is a game rather than a gamified learning system, the design might be more focused on mechanics than content and it may limit the way in which the content can be presented, additionally the reach could be limited since a game may demand high computing power which in turns requires hardware that is not ubiquitous across students. The game seems intended for practice rather than teaching since there was a component of theoretical teaching outside the game as part of the whole learning process. The research does not specify whether the game was designed following pedagogical principles beyond some neuro-linguistical programing (NLP) practices, it is limited to a single scenario, it is not clear whether it requires a client installation or if it can be played in a web browser (this limits the reach). Moreover, the game has been designed to

teach a technical standard but the way it teaches how to identify the right characters to interview may lead the student to a false sense of accuracy since the system gives points for making the right choices and takes points for making the wrong ones, a support that does not exist in real life and thus the program, rather than teaching how to identify stakeholders, might be teaching how to get a high score instead.

3 Methods

Teaching requirements engineering is a human issue rather than a natural phenomenon. Learning for a student involves developing the capacity to understand a concept, use a tool or perform a task. Since observing learning in an objective manner can be challenging, this research could assume an anti-positivist approach, however since its goal has been stated as "to design a gamified system of requirements engineering skills training" the implication is that an artifact must be created and tested, therefore design science research (DSR) is the methodology of choice to attain the objective.

3.1 Design Science Research

The aim of design science research is to solve human issues by means of artifact creation to gain a deeper understanding of the problems addressed (Hevner and Chatterjee, 2010). Peffers et al (2007) have proposed a design science research methodology (DSRM) that is suitable for the present academic work since it is intended for information systems. The main activities suggested by the methodology are:

1. Problem identification & motivation
2. Definition of objectives for solution
3. Design & development
4. Demonstration
5. Evaluation
6. Communication

Activities 1 and 2 have so far been addressed in the current academic work. Teaching requirements engineering is an important task for organizations because it promotes the

adequate use of material and human resources, preventing monetary loss. However, RE is sometimes overviewed as an unessential subject by software engineering students. Gamification has been proved to increase engagement in some studies. Narrative elements and a pedagogical foundation for the design of learning systems could assist with the attainment of learning outcomes. For these reasons, this research proposes to create an artifact that incorporates Merrill's principles of learning, mastery learning, and narrative theory game elements for plot building and gamification.

The artifact will be a web application prototype with minimal content and basic functionality, it shall be exposed to potential end users and their feedback will be gathered to evaluate the impact of narrative gamification on the usability of a learning system. The stated objective of this research is to design a gamified system of requirements engineering skills training. It is assumed that since this system is for learning, there is at least one trainer, professor or class manager who will manage the students and their outputs. The class manager will be able to upload and edit educational content, decide on what actions can be taken by the students and review student input to customize learning where possible.

3.2 Design Process

From the literature review it was inferred that Merrill's principles of learning are a core pedagogical practice that should be implemented as part of the structure of the system. Since Merrill's principles are presented as stages of a learning process, a model was developed where the first 4 principles are a step in the structure of the system. The 5th step will include the narrative design derived from the literature research as the incorporation of the knowledge in the student's world.

Mastery learning has been interpreted as the practice of not allowing the student to advance in the learning cycle until she has fully understood the concept or skill that is intended for learning, thus the intent of the design of the system is to lead the student to repeat the steps derived from Merrill until she proves her knowledge. For this purpose, the 5th step of the

process evaluates the knowledge of the student in the context of a scenario requiring the student to answer questions or make choices related to narrative scenes. In addition, the content administrator may require the student to submit a task uploading a file that the instructor may review. From the process of 5 steps a list of actors and user stories has been drafted to determine the requirements necessary for the system design.

3.2.1 Actors

Two main types of actors have been identified for this platform:

- *Content Administrator:* The administrator is a user capable of uploading and editing content on the platform as well as adding, editing, or deleting other user accounts.
- *Student:* It is a user that can interact with the content and go through the process of navigating the system to learn new concepts and develop new skills.

The content administrator role is intended for a content manager or instructor in charge of uploading and curating learning materials as well as evaluating submissions done by the users. The students on the other hand are expected to be software engineering university students that speak English as their main education language and have access to standard learning hardware such as a laptop or a smartphone.

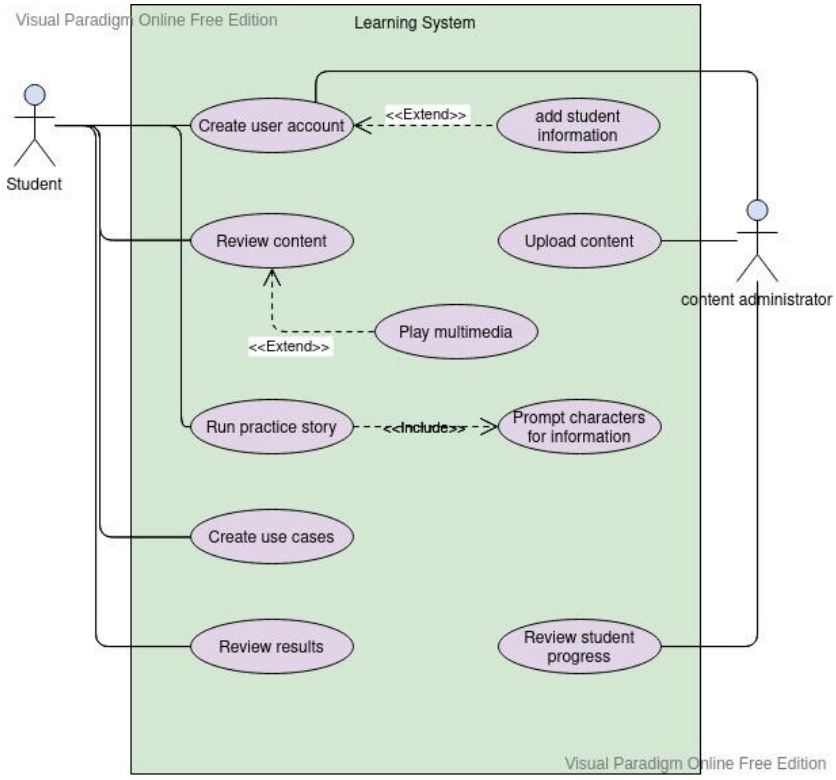


Figure 1 Use case diagram

The student and content administrator should interact indirectly through the application (see figure 1). It is assumed that both actors belong to the same organization with the objective to improve the qualifications of a student with the guidance, skills or knowledge of a content administrator and under his supervision.

3.2.2 User stories

User stories are a general approach to capture and declare functional requirements from the point of view of the user. This approach is used in agile methodologies to express with ease and clarity the features that must be developed (Koelsch, 2016). User stories follow the template "as a (user) I want to (activity) to (objective). An often-used approach to ensure the quality of the user stories is to apply the concept of INVEST, a framework that stands for 6 characteristics that user stories should adhere to: Independent, Negotiable, Valuable, Estimatable, Small and Testable. Researchers have found that the application of INVEST principles to user stories drives self-perceived positive outcomes of productivity and quality for teams that use it, especially for inexperienced ones (Lucassen *et al.*, 2016). The user stories adhere partially to the INVEST framework due to the context of the creation of the

artifact. The application of the negotiable principle is irrelevant since there are no individuals to negotiate with. Moreover, a few stories cannot be considered independent since they rely on the existence of content and therefore cannot be developed in the vacuum of no previous stories implemented. Aside from those 2 exceptions, all the stories comply with the remaining components of the INVEST framework.

For the design process of the artifact, user stories have been chosen as a step to capture the user requirements because their language and simplicity allow them to connect functionality with purpose. The process of redaction has been iterative, formulating early drafts that were later improved as the FURPS+, user flow and mockups that will be created to visualize the implications of each story.

ID of user story	User story
S1	As an instructor I want to upload and modify content so that students can learn from them.
S2	As a student I want to pick up my studies where I left so that I can continue without hassle.
S3	As an instructor I want my content to be presented to the students following Merrill's principles of learning so that they learn effectively.
S4	As a student I want the system to be easy to navigate so that it is clear for me what I am supposed to do, learning without frustration.
S5	As a student I want the learning content to be presented as a story so that learning is engaging and interesting for me.
S6	As an instructor I want the system to ask questions to the student and in the end

	present scenarios based on the answers provided so that the student understands the implications of his choices.

Table 2 Use cases

User stories have been redacted based on the objective and research questions aiming at implementing an online learning system that could incorporate Merrill's principles of learning and storytelling. Stories S1, S2 and S4, (see table 2) have been stated as the pillar to build a learning system, they specifically address the objective (O).

S3 is intended to lead the artifact to answer RQ 3 since the explicit implementation of pedagogical practices is one potential way to address the issue stated. S5 and S6 were stated with the intention to address RQ 1.

It should be noted that additional features not covered by the user stories shall be implemented, however, the stories listed in table 2 represent the core guidelines to design the system, and since the design process is set to be partially iterative, the artifact is meant to be adjusted with end-user feedback.

3.2.3 System Requirements

The FURPS+ system (Grady, 1992) (Eeles, 2001) for classification of requirements engineering ensures a comprehensive coverage or requirements for information systems projects, it addresses the main points to be considered when determining what should be the architecturally significant characteristics of the artifact. The acronym stands for functionality, usability, reliability, performance and supportability with the "+" sign referring to the aspects of design, implementation, interface and also to physical requirements.

Dimension	***Requirement***
Functionality	- An administrator should be able to upload and edit content

	- The system must display learning content related to RE to end-users - The content must be presented in a way that will be consistent with mastery learning and Merrill's principles of learning - The system should test the knowledge of the end-user after the learning has been completed - User accounts should be used to allow users to have asynchronous learning (stop when they need, continue when they want) within a given deadline.
Usability	- The system should be presented in a way that displays a plot, or storytelling elements such as characters fulfilling roles, scenarios or scenes, conundrums or problem solving and/or narrative tension - A user should be able to interact with the prototype with a web browser
Reliability	- The system should be available at the time of user testing
Performance	- The system should run and respond to end-user queries within seconds, not minutes
Supportability	- The system should support the running of automated tests - The system should not rely on proprietary third-party technology
Design requirements	- The user interface should be simple and allow for an easy understanding of what is expected of the end user.
Implementation requirements	- The system must be built using Python 3.x
Interface requirements	- The system should support the presentation and reproduction of multiple content types such as video, audio, text, animated gifs, and images.

- The system should be usable in most available devices and browsers such as desktop computers, laptops, tablets, smartphones with no restriction of vendor.

Table 3 FURPS+ Requirements

The requirements gathered will be used as inputs for decision making related to the final architecture of the artifact, they have been reviewed and collected to ensure a robust approach in design and to ensure the adequate prioritization of resources when building the artifact. Functional requirements were mainly derived from the early versions of user stories, while the usability requirements relate generally to the topic and intent of the artifact. The reliability requirement is limited in scope since the system will be tested to explore features rather than exposed to large numbers of concurrent users for daily use. For the same reason, performance has a low priority, and it is limited to the strictly necessary for research. The supportability requirements have been established aiming at facilitating the reusability of the code and reproducibility of the research without limitation. The implementation requirement is derived from the fact that Python 2 has reached the end-of-life stage since it's last update was released in April 2020. Lastly, the interface requirements were aimed at emulating the conditions that deployment and interaction with users may demand from the artifact.

Since the artifact is intended to answer research questions and not for large scale organizational deployment, the artifact to be developed shall be a prototype with limited capability. Should the artifact be considered for a full organizational deployment, the FURPS+ requirements would need to be amended and from there, the architecture would demand adjustments to ensure adequate operation.

3.2.4 User Flow

A user flow is a flow diagram that showcases the actions and interactions that a user has with a system, describing it as a series of steps. For the purpose of this research, the user flow diagram represents the screens derived from the user stories and the order in which the user is meant to interact with them.

An online learning system's interface can be designed to present the content to students through a menu-based navigation or with a flow-based navigation. In a menu-based system, the user must navigate with a menu to reach the tasks intended such as reading, watching a video or presenting a test. One example of a learning system with menu-based navigation is Moodle, a learning management system broadly adopted by educational institutions. In a flow based- navigation system, the content is mostly presented to the user straight away in a concrete preset order, there might be a menu available but most of the content is presented with little or no use of the menu to find the task to be performed by the student. An example of flow-based navigation system is Duolingo, a mobile and web-based application intended for language learning. Researchers in the field have found that flow navigation is better suited to present content in the context of educational settings since it influences user perception and student performance load, leading to a higher satisfaction and intent of use on the side of the user (Tang *et al.*, 2022).

Based on the requirements, some activities were executed to determine the structure of the system. A user flow from the point of view of the student was established following the pattern of 4 Merrill's steps + 1 narrative combined step. Therefore, the system has been designed to allow the content administrator to create lessons for a student based on the Merrill flow (Appendix 3) using a pattern that compels the instructor to create learning sections.

Merrill states that learners must be engaged in solving real issues, by reviewing the corollaries to the principle this was understood as a need for the instructor to present relevant content to the user. For this purpose, the unit of learning "skill or concept" is conceived and established in the user flow as a "content learning" screen, from here a step considered as a lesson would deliver the content in a format consistent with the need of the instructor. The first lesson of the course should introduce the context of the knowledge to be learned (i.e.: a problem to be solved) but the content would rely on the instructor's intent, knowledge and discretion rather than compel him to present content with strictly specific characteristics.

For the second of Merrill's principles, which states that it is necessary to activate existing knowledge, it is assumed that the content administrator might ask questions or propose thought experiments to the student. This is again a need to present content with the added feature to collect an input from the student whether it is a file or just text from an input field.

For this step a second lesson could be planned, meant to lead the student to interpret the context with her existing personal experience and/or imagination.

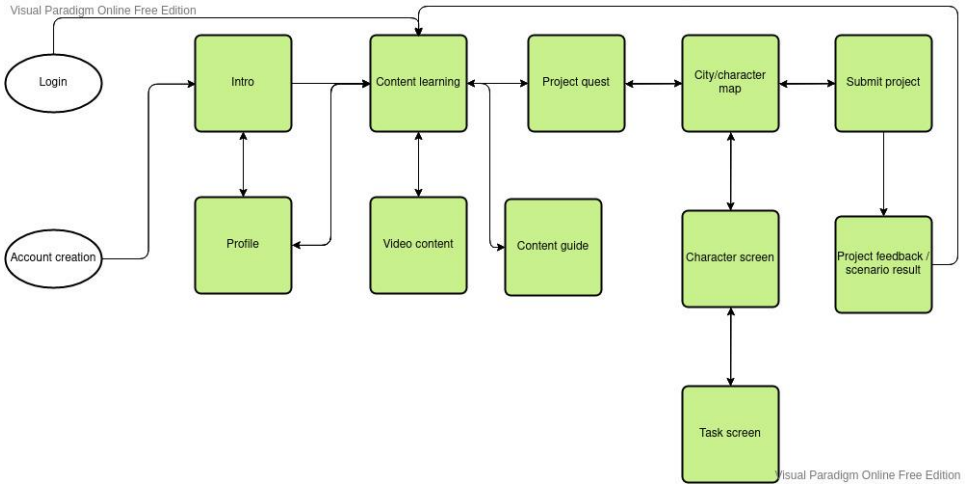


Figure 2 Student user flow

The third principle can be roughly resumed as a step where the instructor must explain the concept or detail how the skill is meant to be performed to the student, in other words the skill must be demonstrated or shown. This leads to the need to show content to the user but since a skill might be a practical issue, a step in the user flow intended to address this step should certainly be able to display multimedia content created or curated by the instructor.

In the fourth step, the student must replicate or apply the knowledge learned as a way of reinforcement. The implication is that the student probably will need to execute a task set by the instructor and then submit it, whether answering a quiz, uploading a file with exercises or sharing a link from a video made by the student for example. Thus, a lesson can be planned for this purpose.

An early conclusion is established then, the student must iterate through lessons that deliver content. Each screen might have slightly different features, but they must adapt mostly to the requirements of the instructor rather than the other way around following the logic of the user stories.

Narrative theory element	*Equivalent step in flow*
Agents	Character screen
Events	Task screen
Objects	Submit project screen
World	Map screen

Table 4 Narrative element equivalence in user flow

For the fifth principle of learning, students must incorporate the gained knowledge into their world. This has been interpreted as applying the knowledge acquired ideally in the real world or alternatively in a simulated scenario. This presents the opportunity to introduce the storytelling design elements into the artifact to deliver the simulated ambience for the student to exercise her new knowledge. Following the combination of narrative theory and the concept of gamification, some of the elements described by narrative theory should be included in the flow to give the system a gameful design, therefore the 4 core elements will be then introduced as screens or subcomponents within screens that compose the flow.

3.2.5 Mockups

As a bridge to connect user flow with user interface the flow screens have been translated into a set of mockup screens that were represented in Figma, a graphic design SaaS tool, to facilitate the visualization of the individual elements to be created in the artifact, using placeholder, non-interactive content. Mockups serve as a representation of the system before the incorporation of the business logic, they allow to identify potential vacuums of description or quantity of the user stories.

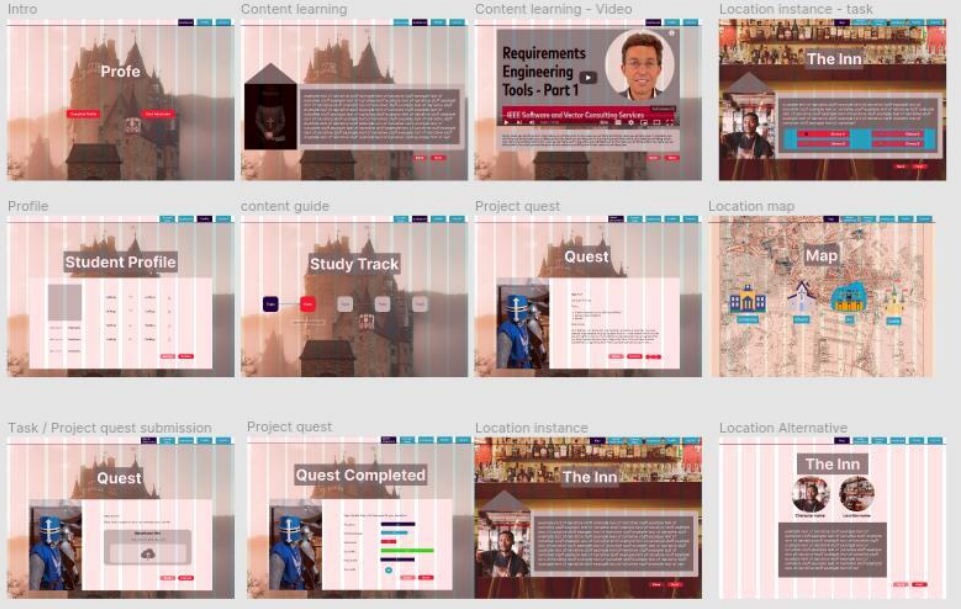


Figure 3 Mockups

For the artifact of this research, 12 user mockups have been designed. This process helps to visualize the incorporation of the narrative elements such as characters, locations, tasks and project submission. Inspiration for the interface was drawn from standard role-playing video games, where it is common to follow the story of the player by seeing pictures of the scenery or characters while displaying boxes of text and answering questions with choices. The map screen, which would allow the student to visit locations where characters could be found, was modeled as a screen with icons with labels representing the individual locations that could be visited. The content presentation screens are planned to be boxes of elements which could contain the learning material: either text or an embedded video.

3.2.6 Domain Model

With a clear vision of the screens and the tasks to be executed by the users, a domain model (see annex 5) has been created in Visio, a diagraming software, to translate interface elements to objects. The classes intended for the four initial steps would include the skills to be learned, the content to be presented, the interactions of the user with the content, the tasks set to be performed and the narrative elements. The actors, events, objects and world have

been accounted for as steps to be taken on screens, but the precise objects must be planned. Actors will be understood as the class characters who will display dialogues which will be shown in a scene. The scene will fulfil the role of events (from the narrative element point of view) to drive the story plot. The world will be set up as a "map" containing "locations" where the "scenes" will happen. The objects will be limited to "submissions", the content administrator can include additional make-believe objects[2] as part of the dialogues.

Since the intention during the narrative step of the learning process is that the student applies the knowledge in a simulated environment, the mechanism designed to fulfill this role will be the interaction of the classes choices, answers and scenario. Choices will represent actions or questions that the student may take when visiting a character, each choice will lead to a reply or reaction from the character and will be worth a score. Answers will record the selection from the student, and it will store the score from the choice selected. Once the student has visited all locations and interacted with all the characters the system will calculate a final score based on the answers and will present a scenario corresponding to the score attained by the student.

All content and objects will be meant to be created by the content administrator, who will fully tailor the information to the skill or concept to be taught, so if an administrator creates a course, he will be able to upload the content, create the characters, write the dialogues, set the choices with their awarded scores and plan the scenarios with their target scores. Students are not intended to be aware of the scores, the feedback received will be in the form of the final narrative scenes.

3.3 Demonstration

As established by the Furps+ requirements, the python programming language has been chosen as the medium to create the artifact. The Django framework from Python was selected as the tool to develop the backend. To avoid duplicity of effort and to speed up development, an existing open source[3] project, Studynet created by Stein Ove, was selected

[2] here objects should be understood as objects from narrative theory, not as objects from object-oriented programming.

[3] Source code available at: https://github.com/SteinOveHelset/studynet_django

as the basis to develop the features required and intended to be tested (namely, the pedagogical structure and the narrative elements of gamification).

Studynet is a learning management system programmed in python that serves courses and lessons containing content which can accept comments, track the completion of tasks and track the courses where the student has enrolled.

The complete process of software development has been executed in a timeline of 2 weeks for design and 3 weeks for programming. The software development process is a hybrid between iterative and cascading project management style. Features were built with little deviation from the planned use cases registered in Jira, however additional features were added during the development of the software to address gaps between the design and the intended objectives. naming convention changes took place regarding the labeling of the classes; therefore, the class names of the final artifact do not reflect the ones in the domain model in a 1:1 proportion. Additionally, the narrative step of the user flow has been denominated as a “quest”, developed as a separate Django application and given its separate set of classes.

3.3.1 System Architecture

A backend has been developed using the REST API framework for Django. A frontend has been built using the VUE 3 JavaScript framework. In Django a model is a class that interacts and represents the objects to be stored in the database. The models that already existed in the open-source project were adjusted to fit the needs of the system and additional models were added to cover the planned features. One such example was the preexistence of the model “Course” which was used as a substitute of the intended “Skill”.

The database used for the artifact on the design face is SQLite, however with the flexibility of Django the prototype can also be set up to work with PostgreSQL for deployment if required. Most of the codebase existing from the original open-source project was utilized to deliver the content, however it had to be extensively modified to adapt for the learning steps planned and the narrative step. In an early iteration of the artifact, the long text form

meant to be delivered as part of the content uploaded by the instructor would not be displayed including the formatting given, this was because the TextField model of Django does not support rich text natively. A third-party application, Ckeditor has been added to the base installation to import a model that would support enriched text.

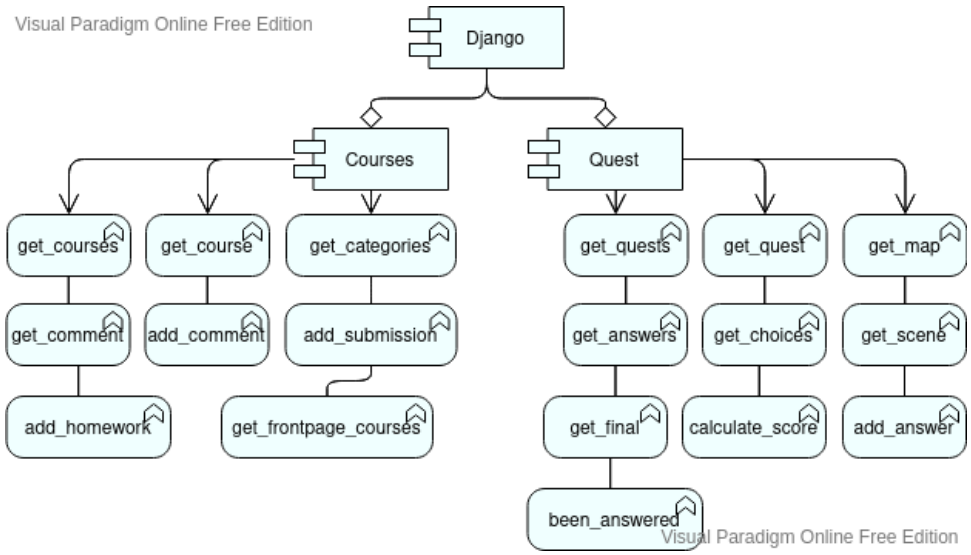


Figure 4 - Available functions for API calls

The backend has 18 views that interact with the models to retrieve or modify data. These views become accessible for the frontend with the API through GET and POST requests. The frontend makes the requests using Axios, an HTTP client for JavaScript. The frontend delivers 12 Vue views based on the mockups complemented with 11 components to display elements like lesson content, comments, choice selections in the narrative step and additional navigation or design components.

The user creation and authentication rely on the preexisting Django auth models, but the system is not currently configured to encrypt the information stored or to communicate with the client using HTTPS. No email verification is required but the input fields of the user account creation screen do validate the email address and the password inputted to ensure that the address has an email structure, and that the password is neither too common nor too

short or easy to guess. No captcha is enabled; therefore, the platform might be vulnerable to brute force attacks.

The version control of the artifact has been performed with a local git repository with daily commits to an online clone based in GitHub, a SaaS code repository. 2 branches were used, a Dev branch for the development of features and a Main branch for the merging once the features had been designed and tried.

3.3.2 Storytelling

The narrative elements in the 5th step of learning (the quest) are intended to present the student with choices that are evaluated based on the knowledge learned in the preceding lessons. The content administrator creates locations, characters, and dialogues intended to present the student with a sense of being submerged in a story while the questions asked, and the answers offered are intended to present a barrier that should both give a sense of participation and narrative tension. The administrator must create potential finals, that is narrative endings to the full quest based in scenarios arising as outcomes from the choices made by the student.

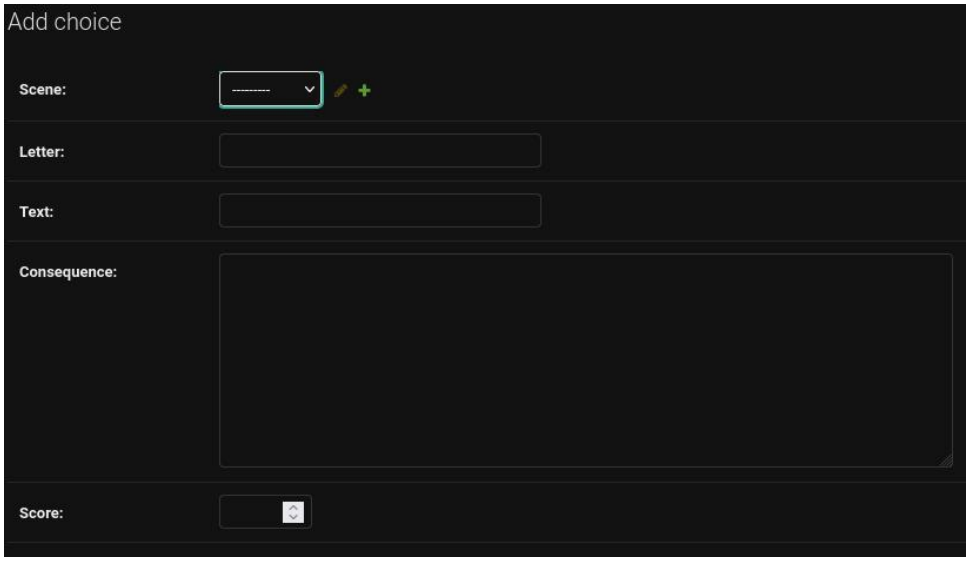


Figure 5 Content administrator panel - choice input fields

The administrator sets the target scores of each potential final and the points awarded for the answers provided to each question. The system calculates the score of the student behind the scenes and when the student has visited all the locations, she is prompted to witness the end of the quest. It must be stated that the system cannot guarantee the quality of the narrative elements produced by the content administrator, such task might require a machine learning model that would fall outside the scope of this research, the system only ensures that the elements created by an instructor will be presented to the student.

3.3.3 Sample Content

To allow for a future test, one full skill course intended to teach how to perform interviews in the context of requirements engineering has been created as sample content for the course, the rest of the platform have 2 additional placeholder courses with no title or content to provide the impression of the presence of additional material, this with the intent of allowing the testing users to have an experience closer to what would be expected on a final full master course with several skills available to learn.

The first lesson, entitled “What Is an Interview?” is comprised of a text taken from Sommerville’s book Software Engineering (2016) describing the concept, the text has a length of 628 words and an estimated reading time of 3 minutes. The second lesson, “Have You Ever Interviewed Anyone?”, is meant to induce the user to reflect on personal story, it presents 1 video on the topic and asks 4 questions to the student. The third lesson, “How to Interview” aims at demonstrating the skill to the student with one video. The fourth lesson “Prepare an Interview” compels the student to design an interview and to submit it either as plain text pasted on an input field or as a word document. When all lessons are done, the student is introduced to the quest.

The quest presents the student with a written narrative introduction giving her the role of someone who has been recently hired in a company. There, she is expected to start gathering requirements from different actors. The map screen displays 2 locations: The meeting room and the hall. In the meeting room a female character from human resources interacts with the student offering 4 options to choose. Likewise, in the hall, a male sales employee is available for interaction. Most of the options available to the student are distractions or offer little value but each scene has at least one choice that could help the user to achieve her goal. Three final scenes are available to be presented depending on the combination of choices made by the student, a failure ending where the user gathered little or no information, an acceptable ending where the student obtained some valuable insights and a successful ending where the student gathered the right data.

3.4 Evaluation

To evaluate the validity of the artifact, the system shall be tested by MSc. software engineering students following the thinking aloud method to examine their interaction with the software during a usability test (Güss, 2018). The users will be briefed on the purpose of the software, they will be given a list of tasks to complete, and they will be instructed to verbalize orally their thinking process when interacting with the interface of the artifact.

Use case ID	*Task*
1	Create a user account
2	Login
3	Start a course
4	Review all lessons
5	Start the quest
6	Visit all characters
7	End quest

Table 5 Test use cases

The sessions will be recorded (with prior consent from the user to do so) and analyzed afterwards. Additionally, some open-ended questions will be formulated after the thinking aloud session to explore their perceptions regarding the system and its features. Such questions include but are not limited to the following list:

- What is your impression of the system?
- What barriers did you face while interacting with the system? (If any)
- What aspects of the system would you change and how?
- How do you rate this system in comparison with other online learning systems?

The information collected from the session will be used to evaluate whether features implemented have a negative effect on usability or not since negative impact has been stated as a potential proxy measure of the impact of gamification features (Seaborn and Fels, 2015). Additionally, the user feedback will be analyzed and used to identify features and improvements necessary for the artifact's design to produce an enhanced version closer to fulfilling the intended purpose of the research.

4 Results

Upon completion of the artifact, to conduct a usability test, users had to be profiled, recruited, briefed, exposed to the platform, and debriefed, then the information was analyzed to discover potential findings. Since the content could be successfully uploaded and ordered in the artifact, given the heavy focus of the design process around the student flow, the decision was made to test the student role. Because the platform and content's aim is to train students in the knowledge of requirements engineering, 2 students of master's degree level in software engineering were recruited from Lappeenranta-Lahti University of Technology in Finland. Both students had already undergone requirements engineering training as part of their education and both had used online learning systems to study the subject, one of the students was male and the other was female.

4.1 Setup

Each test was conducted separately at various times and without the participants establishing contact with one another. The setup used for each test was a remote call using Zoom, an online meeting software, and Anydesk, a tool that allows the user to interact with a remote screen. Zoom was used to record the audio for later analysis. Users were verbally briefed on the purpose of the system, the brief was an introduction to the platform and the research, explaining the thinking aloud methodology. The participants were told what was expected from them verbally, they were also provided with a written set of tasks to perform and instructed to start the test. These tasks related to the use cases to execute: Create an account and login, review all the lessons, and do the quest until finished. The condition to finish the test was the completion of all the tasks.

4.2 Testing

Occasionally during the tests, users were reminded to describe their thinking process aloud when they spent more than 2 minutes interacting without speaking. It took on average 14

minutes for users to execute all use cases. After the tests were run, the users were allowed to speak freely and then they were asked questions related to their interactions.

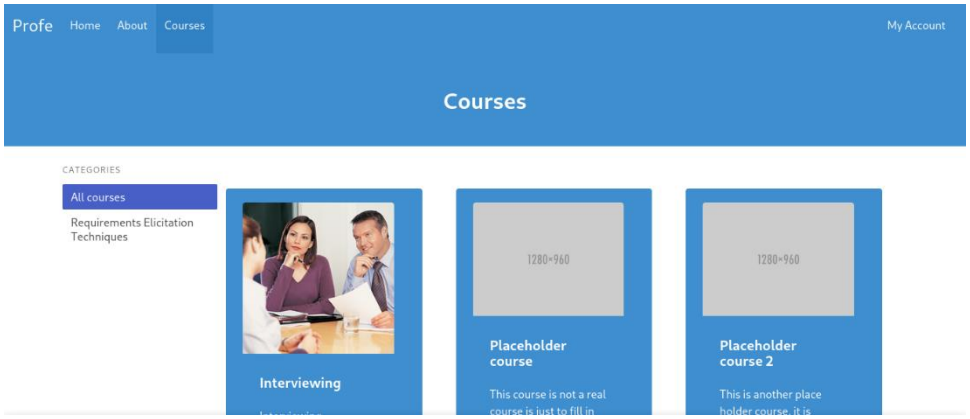


Figure 6 Artifact course selection screen

Subject one expressed curiosity about the platform during the early minutes of the test, commenting "*this looks interesting*", however he faced some navigability issues made evident when he asked "where are the options?... I'm lost" and later stating "what can I do here?" and "why this option doesn't work? I'm confused... I don't know what to do". The last quote was recorded while trying to start the course but being unable to find the button. Later during the test, he also made a remark after reviewing some of the content: "ok I'm interested, how can I interview someone else?" His apparent expectation was interpreted as having the intention to practice the skill right away with someone else using the platform, something the platform was not intended or programmed to do.

Subject two faced incidents that were later used to drive changes in the artifact. One was an individual comment during the thinking aloud test. The subject commented that "*there is a lot of text*." This perception was interpreted as originating from the fact that the text from the first lesson was not properly formatted, therefore the paragraphs were hard to distinguish and read since they were not properly separated. One more comment, *"can I do anything else here?"* mentioned while visiting the dashboard/user profile can be interpreted as the subject expecting additional content or features beyond those presented. Moreover, the subject mentioned at some point "*I have done the course and I have done the quest*" while

the quest had not in fact been finished yet. This led to identifying a lack of visual hints to tell the user that the task had not been totally executed. One question, "*where is the button to go back*?" asked when the user left quest and wanted to return to it helped to address yet another navigation issue derived from the lack of a routing link and button in the categories view.

Initial observations, later corroborated by the aftermath interview, made evident some usability problems derived from lack of visibility of some navigational elements such as the placement, shape and color of the button to start the course, the menu to navigate the lessons, the logout button placement, and the display of long format text in one specific lesson. One of the features (the capacity to comment on the contents of a lesson) was not available across all types of lessons, this derived self-reported feelings of confusion since users had no clear notion of what was expected of them regarding the use of the comments feature, a feature originally taken with no modification from the open-source base code. Additionally, regarding the quest scenario one user reported navigation difficulties due to a lack of visual feedback when selecting the options that each scenario provided confirming the observation noted from the thinking aloud method.

4.3 Analysis

With the tests already executed, the remarks from the answers of the testers and their feedback provided were written down. The most relevant bits of audio from the recording of the tests were transcribed, that is, complete phrases from which some conclusion could be inferred. The notes were clustered, codified and categorized into types of issues associated with the user tasks meant to amend the artifact following the practice of coding from grounded theory (Urquhart, Lehmann and Myers, 2009). Among the notes no feedback was provided that could be associated with challenges or issues with the narrative elements like the characters, locations, scenes, or submissions or the structure of the lessons like preferring all the content to be displayed on a single page, or facing unreasonable expectations regarding the use case tasks.

4.4 Findings

Neither observation during the test nor tester provided feedback reported negative o detrimental interactions derived from the inclusion of a lesson structure that follows pedagogical principles or from the narrative elements incorporated in the system. All use cases were carried out and all the difficulties registered from the testing were associated with features and elements of the navigation, content, existing features, and visibility of interface elements.

Issue	*Type of Issue*	*Solution*
Users prefer an easy and straightforward way to start learning	Navigation	Including a button in the dashboard to start learning right away
Users prefer clarity of the purpose of the platform	Content	Making the objective of the platform more obvious (in the front/landing page)
Users do not understand what answer they chose in a scene	Visual elements	Changing the color of the character choice option
Users visited the same location more times than needed	Visual elements	Changing the color of the button of a location already visited
Users favor consistency in the ability to comment content	Feature	Making the comment section consistent across lesson types
Users favor a clear way to pick up the quest where they left	Feature	Making it easier to continue the quest later if it has been abandoned
Users prefer the ability to log-out at any moment during their study	Navigation	Moving the logout button to a place where it is easier to find

Users skip reading text if they perceive it as hard to read	Content	Declutter the text, give it format

Table 6 Findings

The information gathered from the test was condensed into a list of 8 individual changes to be carried out to overcome the issues reported (see table 6). None of the observations, notes or answers demonstrated a negative effect from the inclusion of the pedagogical structure of Merrill's principles or from the narrative theory elements added to the artifact.

5 Discussion

As stated in its methods chapter, the methodology followed in this research has been established as design science, specifically adhering to the steps described by Peffers et al (DSRM). A learning system has been developed, gameful elements have been included into it and content to teach a skill from the field of RE has been curated to attain the objective laid down by this thesis thus addressing the issues and risks induced by inadequate RE training for the organizations.

After the testing with users, it is evident that the inclusion of the features described above has had no negative impact on the execution of the use cases tested. Users neither expressed excitement nor frustration when interacting with the narrative elements. One unexpected outcome is that one tester described the artifact as “more simple and easier to use” when comparing it with Moodle. Feedback regarding the visibility of navigation items (such as enrollment and logout buttons) also hint at the notion that software engineering students value a simple and intuitive user experience when interacting with learning systems.

The inclusion of narrative structures as a form of gameful design elements for platforms is an unexplored field. A meaningful share of the available literature has focused on additions that increase the sense of competition, gratification, and achievement with features such as experience bars and badges, but no author has been found to use strict literature structures as a feature of gamification. Nevertheless, there are examples of systems that have included elements of roleplaying. One such system is Requengin (García *et al.*, 2020), which might serve as a benchmark of sorts for this research. It should be noted that Requengin is not a gamified system but a serious game as per the concepts explored by Deterding et *al* (2011), nevertheless it has a similar objective since it intends to build a platform to reinforce the understanding and use of the process followed by the ISO/IEC/IEEE 29148:2011 standard and some additional RE knowledge albeit with a different approach. Requengin offers an experience that is immersive and is a simulation with an aesthetical aspect reminiscent of Japanese style animation and videogames. However, Requengin seems to be used mostly as a practice platform to apply, reinforce, and evaluate already acquired skills and concepts rather than to learn new skills. One limitation of the approach includes the difficulty to extend the content or add new content, which might limit the possibility to update with the

latest trends, since the content update must be performed by the developer team. One potential challenge may include the difficulty of distribution, since it is a full game with 3D graphics, it might require high end hardware and the installation of specialized software to support it. Another issue arising may be a steep learning curve specifically for users who do not play or enjoy videogames and who are not used to game controls (i.e.: using the keyboard letters to move a character and the mouse to point the field of vision). In this sense, the artifact proposed by this research might offer a few advantages over the existing solution since it appears to be adequate for providing requirements instruction, or at least non-intrusive and just as good or better than a standard learning platform whereas Requengin seems more adequate for the reinforcement of the knowledge and skills for a specific standard of RE.

One potential explanation to the lack of a strong response on the side of the users in regards of the novel features is that all testers selected were already familiar with the topic of requirements engineering since it is a first-year subject at LUT university. This might dampen the intrinsic curiosity to learn when the users consider themselves knowledgeable about the topic which in time could lead to a weaker intent of interaction. Another potential explanation could lie in the lack of a consistent visual scheme of the contents that were used to run the test (I.e.: pictures of characters and locations). Most multimedia used was readily available rather than customized to present a consistent look and feel of the quest components, this matter has not been controlled since it is perceived as a matter of content qualities rather than system features which falls outside the scope of the research.

There is at least one caveat in this research. It cannot address to what extent the individual content administrator narrative skill (or lack of it) may impact the interaction of the end-user with the application. Custom-made content could lead to unpredictable reactions from the users, this makes it hard to generalize the findings. Additionally, the totality of the content was presented in English to students who spoke English as a second language, the conclusions derived from this research are not likely to apply to students with cultural and linguistical backgrounds that differ from the testing conditions. Nevertheless, there might be a sizeable number of students that do study under circumstances like those of the testers, therefore the use of the artifact could be extended to such sets of students.

Considering that stories and storytelling have an evident influence on games and videogames and using the model of narrative theory of games (Aarseth, 2012), it could be argued that

storytelling and plot elements that can be observed in games could be seen as game design elements and thus implemented as a gamification feature of a software. The artifact created includes elements that are common to plots, stories, and videogames such as characters, locations, and dialogues. The model "choice," which describes the alternative answers or decisions that a user can pick when a character asks a question or presents an issue or situation, can be seen as an item that introduces dramatic tension (Tobias, 2012). The model "final," that denotes a potential scenario that the user will face depending on the combination of choices made, is not unlike the features observed in any videogame with multiple endings such as some Japanese Doujin narrative games (Fiadotau, 2019). The presence of characters, locations, scenes and deliverables correspond with the concepts of agents, events, worlds and objects described by Aarseth (2012).

The sample content included in the testing was based on the list of topics exposed by the Swebok (Bourque, Fairley, and IEEE Computer Society, 2014) and further elaborated by Sommerville (2016). The structure of the lessons displayed in the content followed Merrill's principles of learning and some design choices were influenced by the concept of mastery learning. As an online learning system, the artifact can be used in any device, and it only requires a browser to be accessed. Moreover, the artifact has been designed to present content following a structure that is based on observed pedagogical practices. From the point of view of instruction, the artifact allows content administrators to create, adapt, expand, or modify the content and the narrative elements without having to rely on programming or specialists by using an administration portal.

A potential avenue for future research could be to explore the effects of using both systems (Requengin and the artifact) as complementary to each other rather than as contrasted substitutes. Another study could try to generate the content for the quest section of the artifact with artificial intelligence tools. The scoring system used to serve the final scene of the quest to the student could be based on a machine learning model to avoid the need for instructors to add scores to the choices provided.

6 Conclusions

The current academic work has presented an artifact that conveys novel approaches in regards of REE. The system designed has addressed RQ 1 “Can roleplaying and/or storytelling be implemented in an online learning system?” by defining and incorporating Django objects which reflect the structure of an adventure with items that can be observed in archetypical stories and videogames modeled with narrative theory. In the structure of plots, the audience of a plot consciously or unconsciously tries to guess what fate will befall the characters given the circumstances that surround them. Narrative tension arises from barriers that are faced by the users and by the consequences of their choices. In the artifact of this thesis, the students are confronted with circumstances and decisions that will have an array of consequences derived from their answers. Since elements like agents, a world, events and objects have been introduced, it can be therefore considered that the research question has been answered.

The literature reviewed in the related works section sheds some light on a potential answer for RQ 2 “What key skills would be necessary for comprehensive training of requirements engineering for engineering students?”. Normative standards like the ISO/IEC/IEEE 29148:2011 have been the focus of research with a similar objective to this one, nonetheless, the standard makes great focus on a structured approach but makes little reference on how to develop the human skills necessary to face actual organizational challenges. Structure without social tact may diminish the effectiveness of the formats and techniques presented by the standard. The swebok is broad and comprehensive but it lacks depth and delegates the elaboration of the concepts to Sommerville, who in turn is devoid of meaningful details. One of the conclusions of this academic work regarding answering RQ2 is that a comprehensive learning system should probably include a fair share of content inspired by the Swebok complemented with Somerville’s textbook adding the 2018 version of the ISO/IEC/IEEE 29148 standard instead of the 2011 since new and reviewed practices have been included.

This thesis tries to answer RQ3 “How could such a system facilitate the teaching of key skills?” incorporating the pedagogical principles described by Merrill and introducing restrictions in the artifact that resemble the core principle of mastery learning. It achieves

that by not allowing the student to progress to the next concept until she has fully undergone the formative cycle and by sending her back to the lessons if she has failed to achieve the learning outcomes during the quest phase of the course. Moreover, a user flow that reflects the 5 principles of instruction should help the student to learn in structured, purposeful and gradual lessons where she is first introduced to the skill or concept, then prompted to recall relevant experiences, she is shown how to perform the task or taught about the concept in depth, she is asked to perform the skill or demonstrate the knowledge and she is finally guided to incorporate the knowledge in her world by using it in a narrative environment.

The artifact designed during this research could be used by organizations such as universities to train or improve the skill of engineers regarding the topic of requirements engineering. This field is relevant because good requirements engineering saves time, effort and money while bad RE could derail a software engineering project. Further research is required to prove if the artifact can generate a meaningful increase of the effectiveness of the content or of the engagement of students, however the outcome of this study implies that the gamification performed offers no barrier and successfully incorporates design elements akin to those existing in storytelling, a practice known to capture the attention of people, therefore holding the potential to do so similarly for students.

The research performed regarding REE rarely considers pedagogical aspects of skill training and often develops tools, techniques and frameworks devoid of this dimension. The current thesis proposes a way of teaching that is in line with the latest trends of REE identified by Daun et al (2021) as it involves problem solving, role-playing and gamification on top of a pedagogically based structure.

6.1 Future research

The designed artifact has been published under an MIT license in GitHub[4]. Future researchers could attempt to test the system in a different organizational setting such as a business to find out what effect the artifact has on the learning outcomes of specific types of users. If such a study were to be paired with research to find out a relation between RE

[4] see https://github.com/morc4/edu_django

training and project success rates, greater clarity could be obtained in regards of the adequacy of the artifact to succeed in its ultimate purpose.

Additional research can be carried out regarding the degree of gamification that can be obtained by adding or subtracting some narrative theory elements such as world or events. Users could be asked their perceptions on sense of immersion, attraction and other variables. Different strategies to introduce narrative elements inspired in other games can also be tested. Examples could include introducing music or foley effects, animations like changes in the facial expressions of the characters, changes in the location's color palette, new interactive objects like a student notebook or others.

7 References

Aarseth, E. (2012) ‘A narrative theory of games’, in *Proceedings of the International Conference on the Foundations of Digital Games*. New York, NY, USA: Association for Computing Machinery (FDG ’12), pp. 129–133. doi:10.1145/2282338.2282365.

Alami, D. and Dalpiaz, F. (2017) ‘A Gamified Tutorial for Learning About Security Requirements Engineering’, in *2017 IEEE 25th International Requirements Engineering Conference (RE)*. *2017 IEEE 25th International Requirements Engineering Conference (RE)*, Lisbon, Portugal: IEEE, pp. 418–423. doi:10.1109/RE.2017.67.

Bourque, P., Fairley, R.E., and IEEE Computer Society (2014) *Guide to the software engineering body of knowledge*.

Campbell, J. (2008) *The Hero with a Thousand Faces*. New World Library.

Connor, A.M., Buchan, J. and Petrova, K. (2009) ‘Bridging the Research-Practice Gap in Requirements Engineering through Effective Teaching and Peer Learning’, in *2009 Sixth International Conference on Information Technology: New Generations*. *2009 Sixth International Conference on Information Technology: New Generations*, Las Vegas, NV, USA: IEEE, pp. 678–683. doi:10.1109/ITNG.2009.134.

Daun, M., Grubb, A.M. and Tenbergen, B. (2021) ‘A Survey of Instructional Approaches in the Requirements Engineering Education Literature’, in *2021 IEEE 29th International Requirements Engineering Conference (RE)*. *2021 IEEE 29th International Requirements Engineering Conference (RE)*, Notre Dame, IN, USA: IEEE, pp. 257–268. doi:10.1109/RE51729.2021.00030.

Deterding, S. *et al.* (2011) ‘From game design elements to gamefulness: defining “gamification”’, in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. New York, NY, USA: Association for Computing Machinery (MindTrek ’11), pp. 9–15. doi:10.1145/2181037.2181040.

Eeles, P. (2001) ‘Capturing Architectural Requirements’, p. 13.

Fernández, D.M. *et al.* (2017) ‘Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice’, *Empirical Software Engineering*, 22(5), pp. 2298–2338. doi:10.1007/s10664-016-9451-7.

Fiadotau, M., 2019. Indie and Dojin Games: A Multilayered Cross-Cultural Comparison. G*amevironments*, (10), pp.46-46.

García, I. *et al.* (2020) ‘A serious game for teaching the fundamentals of ISO/IEC/IEEE 29148 systems and software engineering – Lifecycle processes – Requirements engineering at undergraduate level’, *Computer Standards & Interfaces*, 67, p. 103377. doi:10.1016/j.csi.2019.103377.

Gibson, J.P. (2000) ‘Formal requirements engineering: learning from the students’, in *Proceedings 2000 Australian Software Engineering Conference*. *Proceedings 2000 Australian Software Engineering Conference*, pp. 171–180. doi:10.1109/ASWEC.2000.844574.

Grady, R.B. (1992a) *Practical software metrics for project management and process improvement*. Englewood Cliffs, NJ: Prentice Hall.

Güss, C.D. (2018) 'What Is Going Through Your Mind? Thinking Aloud as a Method in Cross-Cultural Psychology', *Frontiers in Psychology*, 9, p. 1292. doi:10.3389/fpsyg.2018.01292.

Hevner, A.R. and Chatterjee, S. (2010) *Design research in information systems: theory and practice*. New York ; London: Springer (Integrated series in information systems, v. 22).

Koelsch, G. (2016) *Requirements Writing for System Engineering*. 1st ed. 2016. Berkeley, CA: Apress : Imprint: Apress. doi:10.1007/978-1-4842-2099-3.

Kulik, C.-L.C., Kulik, J.A. and Bangert-Drowns, R.L. (1990) 'Effectiveness of Mastery Learning Programs: A Meta-Analysis', *Review of Educational Research*, 60(2), pp. 265–299. doi:10.3102/00346543060002265.

Landers, R.N. (2014) 'Developing a Theory of Gamified Learning: Linking Serious Games and Gamification of Learning', *Simulation & Gaming*, 45(6), pp. 752–768. doi:10.1177/1046878114563660.

Landers, R.N. *et al.* (2018) 'Gamification Science, Its History and Future: Definitions and a Research Agenda', *Simulation & Gaming*, 49(3), pp. 315–337. doi:10.1177/1046878118774385.

Looyestyn, J. *et al.* (2017) 'Does gamification increase engagement with online programs? A systematic review', *PLOS ONE*. Edited by F. Amblard, 12(3), p. e0173403. doi:10.1371/journal.pone.0173403.

Lucassen, G. et al. (2016) 'The Use and Effectiveness of User Stories in Practice', in Daneva, M. and Pastor, O. (eds) Requirements Engineering: Foundation for Software Quality. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 205–222. doi:10.1007/978-3-319-30282-9_14.

de Marcos-Ortega, L. *et al.* (2020) 'Gamifying Massive Online Courses: Effects on the Social Networks and Course Completion Rates', *Applied Sciences*, 10(20), p. 7065. doi:10.3390/app10207065.

Merrill, M.D. (2002) 'First principles of instruction', *Educational Technology Research and Development*, 50(3), pp. 43–59. doi:10.1007/BF02505024.

Ouhbi, S. *et al.* (2015) 'Requirements engineering education: a systematic mapping study', *Requirements Engineering*, 20(2), pp. 119–138. doi:10.1007/s00766-013-0192-5.

Peffers, K. *et al.* (2007) 'A Design Science Research Methodology for Information Systems Research', *Journal of Management Information Systems*, 24(3), pp. 45–77. doi:10.2753/MIS0742-1222240302.

Ritter, S. *et al.* (2016) 'How Mastery Learning Works at Scale', in *Proceedings of the Third (2016) ACM Conference on Learning @ Scale*. *L@S 2016: Third (2016) ACM Conference on Learning @ Scale*, Edinburgh Scotland UK: ACM, pp. 71–79. doi:10.1145/2876034.2876039.

Sailer, M. and Homner, L. (2020) ‘The Gamification of Learning: a Meta-analysis’, *Educational Psychology Review*, 32(1), pp. 77–112. doi:10.1007/s10648-019-09498-w.

Seaborn, K. and Fels, D.I. (2015) ‘Gamification in theory and action: A survey’, *International Journal of Human-Computer Studies*, 74, pp. 14–31. doi:10.1016/j.ijhcs.2014.09.006.

Sommerville, I. (2016) *Software engineering*. 10. ed., global ed. Boston Munich: Pearson (Always learning).

Soo, M.T. and Aris, H. (2018) ‘Game-Based Learning in Requirements Engineering: An Overview’, in *2018 IEEE Conference on e-Learning, e-Management and e-Services (IC3e). 2018 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, pp. 46–51. doi:10.1109/IC3e.2018.8632650.

Svensson, R.B. and Regnell, B. (2017) ‘Is role playing in Requirements Engineering Education increasing learning outcome?’, *Requirements Engineering*, 22(4), pp. 475–489. doi:10.1007/s00766-016-0248-4.

Tang, P. *et al.* (2022) ‘How information presentation formats influence usage behaviour of course management systems: flow diagram navigation versus menu navigation’, *Behaviour & Information Technology*, 41(2), pp. 383–400. doi:10.1080/0144929X.2020.1813331.

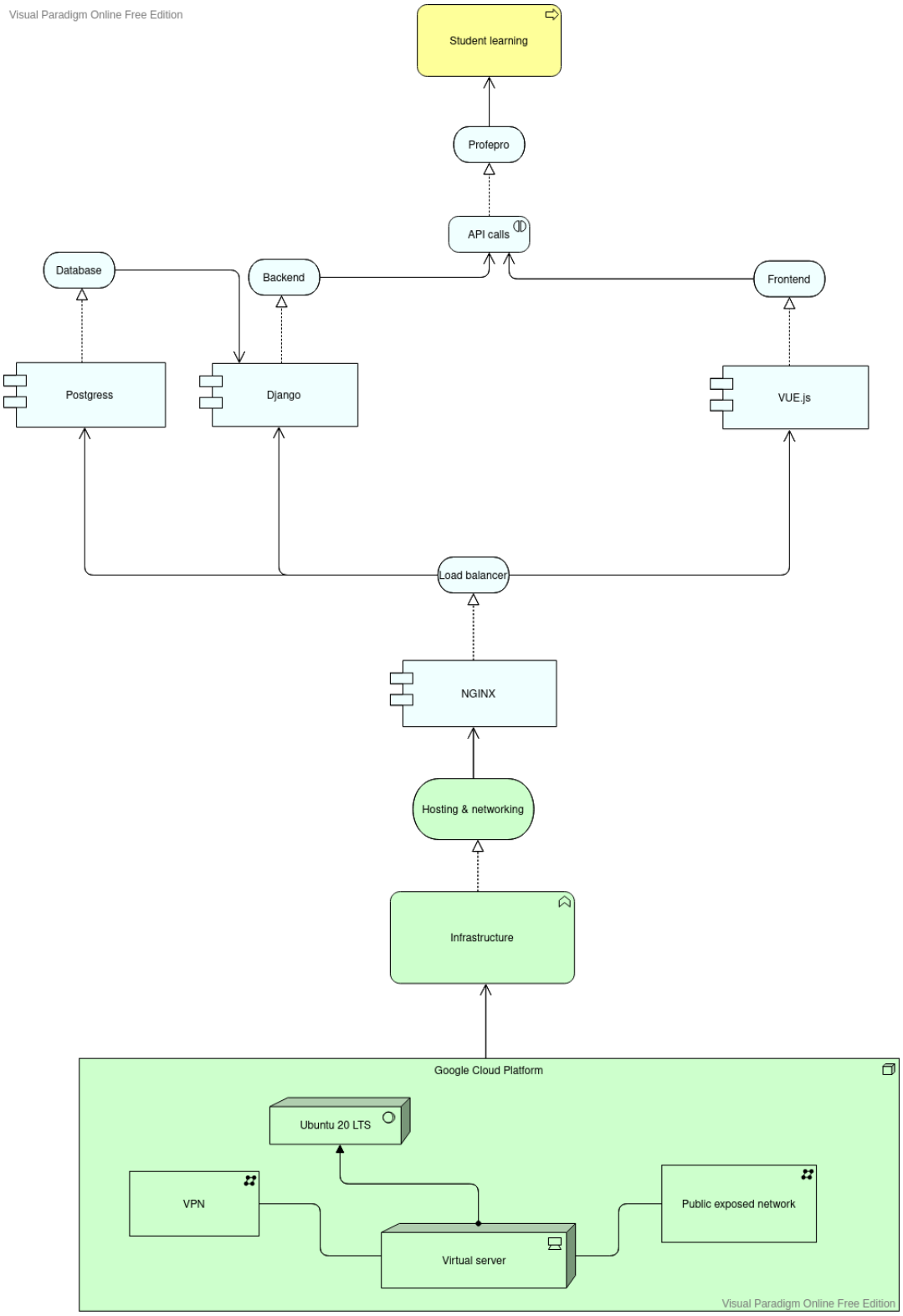
Tobias, R.B. (2012) *20 master plots - and how to build them.*

Toda, A.M. *et al.* (2019) ‘A Taxonomy of Game Elements for Gamification in Educational Contexts: Proposal and Evaluation’, in *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT). 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, pp. 84–88. doi:10.1109/ICALT.2019.00028.

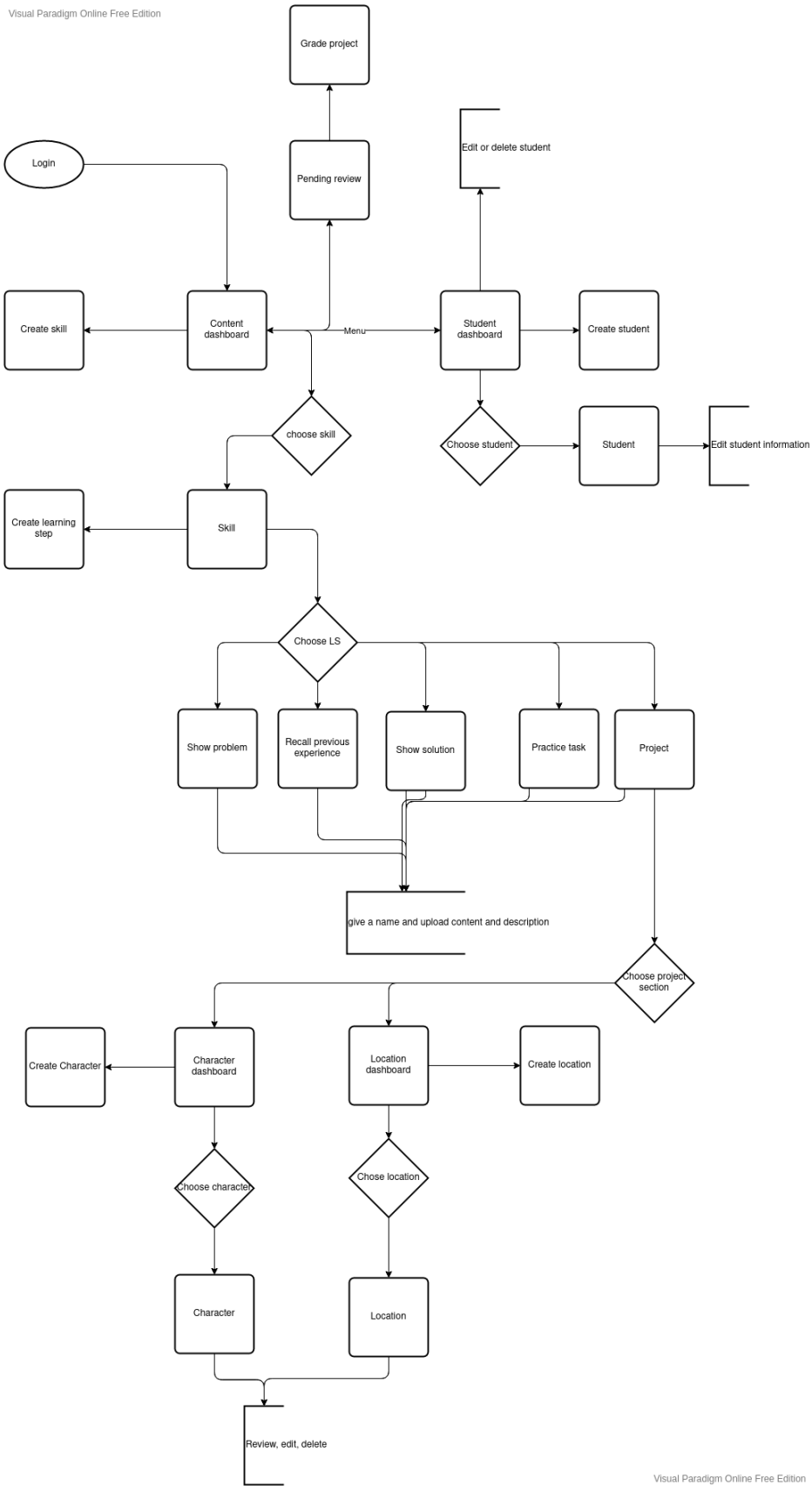
Topîrceanu, A. (2017) ‘Gamified learning: A role-playing approach to increase student in-class motivation’, *Procedia Computer Science*, 112, pp. 41–50. doi:10.1016/j.procs.2017.08.017.

Urquhart, C., Lehmann, H. and Myers, M.D. (2009) ‘Putting the “theory” back into grounded theory: guidelines for grounded theory studies in information systems: Guidelines for grounded theory studies in information systems’, *Information Systems Journal*, 20(4), pp. 357–381. doi:10.1111/j.1365-2575.2009.00328.x.

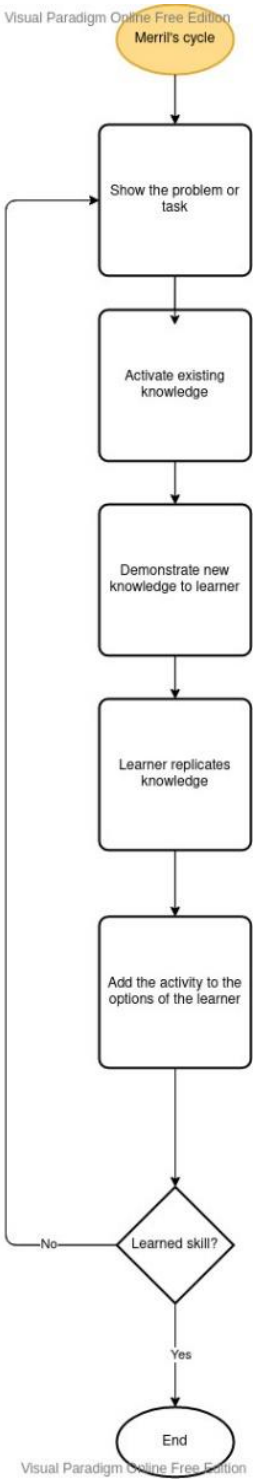
Appendix 1. ArchiMate Diagram.



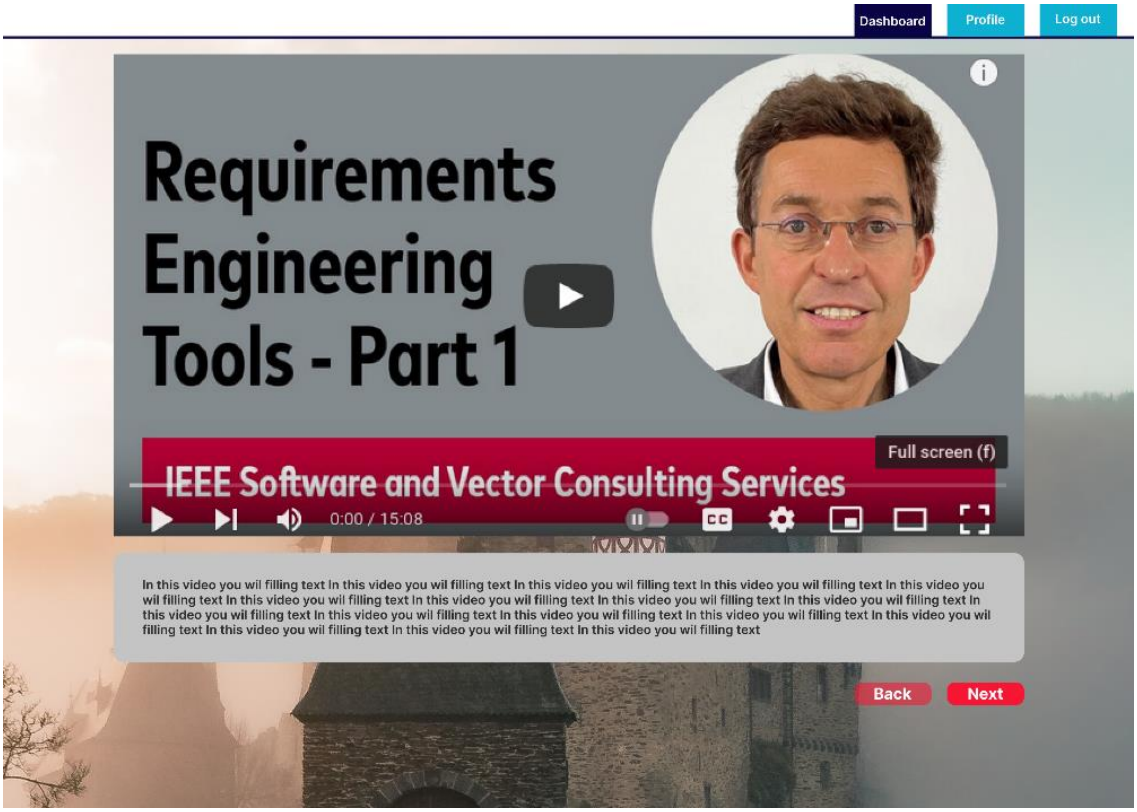
Appendix 2. Instructor User Flow.



Appendix 3. Merrill's Principles as a Flow.



Appendix 4. Student Use Case Mock-up (Close-Up)



Appendix 5. Domain Model