

## Non-functional Requirements

### *Performance Requirements*

**R1.** "The system must perform efficiently and fast despite the number of users." [4]

Moodle, the framework that we are using, has guidelines on how to keep the performance of our system at its best as well as make it possible for it to scale well. These guidelines include having each page have a fixed number of database queries, limiting the amount of RAM each page requires to generate, and being cautious about other external calls<sup>[1]</sup>. Adhering to these guidelines will ensure that our code is able to scale and perform. [Including screenshot examples of code that adheres to these guidelines for code quality assurance purposes]

In order to improve the performance of our code we are going to need to measure it during development, and measure it in production. [Include performance tests during testing]

The Moodle framework is actually very scalable. The largest sites using this platform have up to 1 000 000 users, up to 50 000 courses, up to 5 000 users per course, up to 50 roles, up to 100 course categories nested up to about 10 levels deep<sup>[1]</sup>. Moodle sites can be even larger. Using generator scripts can allow us to test our code in a Moodle site that is not small. [Include a generator script test after each iteration, show a demonstration, or test with larger values and demonstrate too]

Tools: MySQL JOINS and subqueries, JMeter<sup>[4]</sup>,  
<https://github.com/moodlehq/moodle-performance-comparison> , Test site generator<sup>[2]</sup>, Test course generator<sup>[3]</sup>, and if we're using PostGres, we can use a script that can parse the logs and output the top 10 slow queries which is ready to be plugged into a cronjob to email you when the site is in production: [http://git.catalyst.net.nz/gw?p=pgtools.git;a=blob\\_plain;f=scripts/pg-log-process.pl;hb=refs/heads/pg-log-process-multidb](http://git.catalyst.net.nz/gw?p=pgtools.git;a=blob_plain;f=scripts/pg-log-process.pl;hb=refs/heads/pg-log-process-multidb)

All of these should be done during each and every iteration through the project, and relevant documentation should indicate as much. A checklist should accompany the documentation to show that the rules have been observed where screenshots and demonstrations cannot be done.

**R1.1** "The system must limit each web page to a fixed number of database queries."

**R1.2** "The system must limit the amount of RAM each page requires to generate."

**R1.3** "The system must limit the amount of external calls."

**R2.** "The system must be able to work efficiently despite the content linked to in the database."  
[4]

These issues have largely been dealt with in the information presented above (for the first requirement); also performance measures should ideally have performance metrics.

To be amended: Perhaps this is referring to the efficiency of the database. That is should maintain a

constant response even though a large amount of data is stored within it?

### *Quality Requirements*

**R3.** "The system must be available ninety-nine percent of the time." [5]

Both hardware and software faults would need to be considered in order to get measures of reliability, mean time to failure (MTTF), and availability<sup>[5]</sup>. We have set for an availability of ninety nine percent, which means that our system must have an annual downtime of forty four to eighty seven hours<sup>[5]</sup>.

Since the site has not been up, we cannot actively measure these performance parameters. We can however make sure to use tools that will continually monitor the uptime of our site and notify the relevant individuals when the site is down. Using the monitoring service, TOODLE (<http://toodle.org>), we would be able to check the uptime of our site in various intervals using a robot<sup>[6]</sup>. If something wrong is detected, an email will be sent with the issue details as well as a screenshot of the site; this being done every month<sup>[6]</sup>. The tool will also be able to check the uptime of the database<sup>[7]</sup>.

Usage of these statistics (provided from the tool mentioned above) whilst the website is live, will allow us to take note of the system's uptime as well as calculate our MTTF (mean time to failure), MTTR (mean time to repair) and our availability<sup>[5]</sup>.

### *Safety Requirements*

**R4.** "The system must save test progress before the test is submitted by the user in case of a submission failure or loss of internet connection." [2]

To be amended: this can be done using cookies and local storage which can store from 5 to 24 MB of data\*. How would we export this data from Moodle to the user's browser's local storage variables? The issue of different browsers offering different capabilities.

This can also be done by automatically saving the answers in a database until the user returns to the quiz. A deletion policy can be followed to have quizzes that are returned to and submitted, removed from the 'cache' database and deletions to be done automatically after a certain period of time.

**R4.1** "The system must save test progress whilst the user is taking the test."

**R4.1.1** "The system must display that the question has been saved after it is edited."

**R4.1.2** "The system must save the question answer in a cache database."

**R5.** "The system must retrieve saved test progress in the case of a submission failure or loss of Internet connection."

**R5.1** "The system must retrieve the question answer from the cache database."

**R5.2** "The system must display the question answer on the question."

**R5.3** "The system must display that the question has been retrieved."

[Google software safety requirements examples, and these other requirement types, we might be leaving something important out]

### *Security Requirements*

[think, how will we protect the programs or data from malicious attack]

**R6.** "The system must identify the user using their email and password before gaining access to the website and accessing user profiles amongst other features." **[5]**

This is necessary to identify users that have access to the data and or programs of the system. User profile information is then only made available to identified (and authenticated users).

Moodle has built in settings that ensure password complexity and options for reporting login failures which are activatable by the administrator<sup>[8]</sup>. Password complexity enforces certain rules to make sure that the passwords are not too simple e.g. requiring a certain minimum number of characters<sup>[8]</sup>.

Moodle has a lockout system from version 2.5 which could be turned on by following the path "Administration > Site Administration > Security > Site policies... Account lockout threshold<sup>[8]</sup>". It also counts the number of login attempts and can alert the administrator when they are too high (threshold dependant\*).

**R6.1** "The system must capture and check the authenticity of the email."

**R6.1.1** "The system must allow the user to enter their email."

**R6.1.2** "The system must check for an email match in the database."

**R6.1.3** "The system must take note of an incorrect email."

**R6.1.3.1** "The system must alert the user of an incorrect email."

**R6.1.3.2** "The system must increment the login attempts."

**R6.1.3.3** "The system must alert the administrator if the login attempts have met the threshold."

**R6.2** "The system must check if the capture and check the if the password matches the account linked to the email."

**R6.2.1** "The system must allow the user to enter their password."

**R6.2.2** "The system must check for a password match in the database using the email."

**R6.2.2.1** "The system must use the email to isolate the relevant password in the database."

**R6.2.2.2** "The system must check if the password matches the selected one in the database."

**R6.2.3** "The system must take note of an incorrect email and password combination."

**R6.2.3.1** "The system must alert the user of an incorrect email and password combination."

**R6.2.3.2** "The system must increment the login attempts."

**R6.2.3.3** "The system must alert the administrator if the login attempts have met the threshold."

**R6.2.4** "The system must alert the owner of the account of a login attempt."

**R6.3** "The system must check if OTP is enabled for the user profile and use it accordingly."

**R6.3.1** "The system must provide an input field for the OTP."

**R6.3.2** "The system must send the OTP to the user."

**R6.3.2.1** "The system must obtain preferred contact details of the profile owner from the database."

**R6.3.2.2** "The system must generate an OTP."

**R6.3.2.3** "The system must send the OTP to the owner using the preferred contact details."

**R6.3.3** "The system must check if the OTP matches the input from the user."

**R6.3.3.1** "The system must alert the user of a match if it is so."

**R6.3.3.2** "The system must allow the user to try again."

**R6.3.3.2.1** "The system must increment the OTP counter."

**R6.3.3.2.1.1** "The system must check if the OTP counter matches the threshold."

**R6.3.3.2.1.2** "The system must start a timer and halt the user until the time finishes."

**R6.3.3.2.1.3** "The system must alert the administrator."

**R6.3.4** "The system must log in the user to the website."

**R6.3.4.1** "The system must redirect the user to the relevant page."

**R7.** "The system must provide a OTP (One Time Password) for user accounts that have sensitive access rights i.e. admin accounts." **[3]**

To be amended: email or SMS notification or even fingerprint (using mobile device with fingerprint sensor) or QR codes scanned using their mobile (like whatsapp notifications...could browsers do this?)

**R7.1** "The system must enable users to toggle OTP usage for their accounts."

**R7.1.1** "The system must allow users an input field to toggle OTP usage in user profiles."

**R7.1.2** "The system must store the OTP usage setting in the database once set."

**R7.1.3** "The system must store the OTP usage setting in the database once unset."

**R8.** "The system must encrypt the connection between the webserver and the client browser." **[4]**

To be amended: this shall be done through SSL by securing an SSL certificate.  
We can acquire a Free domain-validated certificate from "Let's Encrypt" (<https://letsencrypt.org/>) which makes the process of installing and managing certificates as easy as possible<sup>[9]</sup>.

**R9.** "The system must scan all files that are uploaded to site for malicious software such as viruses, worms, Trojans etc." **[5]**

To be amended: antivirus plugins can be incorporated to scan all uploaded files to ensure immunity from malicious software and thus the security of data and the system's programs.  
Moodle enables the use of an open source antivirus scanner called Clam AV which can be enabled by following the path "Settings > Site Administration > Security > Antivirus<sup>[11]</sup>". ClamAV would need to be installed on the server however<sup>[11]</sup>; <http://www.clamav.net/>.

**R10.** "The system must require the admin to authenticate themselves before deletion, and uploading of resources." **[5]**

**R10.1** "The system must provide an input field for the user to enter their password."

**R10.2** "The system must check if the password matches."

**R10.2.1** "The system must obtain the user's email from session data."

**R10.2.2** "The system must use the user's email to check for a match in the password from the database."

**R10.2.3** "The system must increment the password tries."

**R10.2.4** "The system must check if the password tries match the threshold."

**R10.2.5** "The system must disable the action if password tries are matched."

**R10.2.5.1** "The system must disable the specific upload button."

**R10.2.5.2** "The system must start a timer for a specified period."

**R10.2.5.3** "The system must check if the timer is finished."

**R10.2.5.4** "The system must enable the button after the timer is finished."

**R10.2.6** "The system must alert the admin of the action."

**R10.2.6.1** "The system must send the admin an email containing relevant information."

**R10.3** "The system must scan the uploaded file for malware if it is uploaded."

**R10.4** "The system must delete (physically or logically) the file from the database if it is deleted."

**R11.** "The system must log access to certain resources so that records are kept if users attempt foul play." + {assuming multiple contributors?} [0]

To be amended: once certain pages or resources are accessed, an entry of the user and their behaviour is logged in a database to prevent denial in the case of foul play. Con: this may result in storage overhead, which can be reduced by introducing a deletion policy to delete older, hopefully unrequired records.

Users capable of doing this are those that can upload and delete files, which at this point is the admin. A solution to this is saving each session activity i.e. the date, pages accessed, files accessed, files deleted, files uploaded, comments made in discussion boards, time session ends. Session start time or session end time along with the user id may be used as record identifiers.

**R12.** "The system must encrypt sensitive student information to ensure student privacy and account security." [0]

**R12.1.** "The system must encrypt student passwords." [5]

To be amended: Moodle uses the MD5 algorithm as well as salting to encrypt and secure passwords. This

mechanism will also be used to encrypt students emails, and cell phone numbers, the latter being optionally made available for them for OTP codes (for forgotten passwords and extra login authentication). This could be extra, pointless processing...decrypting the emails before sending notification mail.

**R12.2.** "The system must encrypt students names, emails and cell phone numbers stored in the database." [3]

**R13.** "The system must have contingencies set in place to thwart common website attacks." [2]

Resilience requirements will be specified with the analysis of each attacks}

**R13.1.** "The system must have prepared counter measures to survive a DoS attack." [3]

**R13.2.** "The system must not allow for the possibility of SQL injection." [3]

**R13.3.** "The system must be secured against cross site scripting." [3]

**R13.4** "The system must be secured against local file inclusion attacks." [3]

**R14.** "The system must separately set up backups of the data required by the subsystems in case of a loss or corruption of data." [3]

{code backups should be kept outside as well}{data here including database passwords, third party software authentication information (the passwords and etc) required for interfacing}

**R14.1.** "The system must encrypt sensitive information backups such as software interfacing credentials i.e. usernames and passwords needed to interface with third party software." [1]

**R14.2.** "The system must back up academic data and other databases off the main server/s." [3]

**R15.** "The system must provide alternative ways to access the server incase login through a web page is inaccessible due to an attack." [1]

[In case the server is completely {shut of}...ssh tunneling to access the system? Execute diagnosis remotely, set all of these up...include SMTP servers on the server so that the admin can receive access to the emails, in case Moodle sent the admin an email in the case where a user was trying to brute force entry into the system. Set these up, and include specific requirements. X window etc] {shut off the connection if suspicious commands are being sent through--cut the tcp pipe}+[is this possible?]{Specify versions and standards}

### *Interface Requirements*

**R16.** "The system must have a simple user interface layout that is not as clustered as possible." **[5]**

To be amended: may incorporate Android material design principles. However, prototyping designs and changing them incrementally according to user feedback should work.

**R17.** "The system must have instructional videos to help users navigate and use it." **[2]**

To be amended: this videos may be included in the home page, which is redirected to after the user logs into their account.

**R18.** "The system must have intuitive user interface logos and features." **[4]**

check: i. <https://www.hongkiat.com/blog/defining-designing-intuitive-ui/> ; ii. <https://www.creativebloq.com/web-design/examples-ui-design-7133429> ;

**R19.** "The system must be able to adjust appropriately to the client's screen size without cropping out elements. The pages must make use of responsive design." **[5]**

To be amended: Bootstrap; for all of these, what does Moodle implement?

## **Functional Requirements**

### *Users*

**R21.** "The system must compress data before sending it to users in order to conserve bandwidth, for both the server and the end users." **[Pending]**

This is considering data costs on both ends, in an effort to reduce the data usage for clients as well as that of the server. External services such as YouTube may not be included in this service thus in order to do this, it may make more sense to store the videos in our own database since we have more control over those connections.

**R22.** "The system must allow new users to register if they are new to the website." **[5]**

**R22.1** "The system must allow users to register using social media accounts."

There is an issue here I want to resolve quickly: while profile data can be optioned from these profiles, what about passwords? Hackerrank allowed me log in using my Google profile however when I wanted to delete my profile it requested a password, something I didn't have because of the social media registration. If passwords indeed cannot be obtained, then we must allow users to choose their own.

As is stands, it seems as though users who login using social media can use this obtain all the time for login and authentication purposes and might even need to use a password.



**R22.1.1** "The system must allow users to register using Facebooks accounts."

The main motivation behind this was that students might not have email accounts but have Facebook accounts since Facebook allows them to register using their cellphone numbers.

**R22.1.2** "The system must allow users to register using Google accounts."

**R22.1.3** "The system must allow users registered via social media to choose their own passwords."

**R22.2** "The system must allow new users to register using their email."

**R22.2.1** "The system must obtain a new user's email addresses for registration."

**R22.3** "The system must obtain a new user's username for registration."

**R22.4** "The system must obtain a new user's password for registration."

**R22.5** "The system must obtain a new user's province for registration."

**R22.6** "The system must obtain a new user's grade for registration."

**R22.7** "The system must obtain a new user's date of birth for registration."

**R23.** "The system must allow registered users to login so that they can browse the content." **[5]**

**R23.1** "The system must allow registered users to login using social media accounts."

**R23.1.1** "The system must allow registered users to login using Google accounts."

**R23.1.2** "The system must allow registered users to login using Facebook accounts."

**R23.2** "The system must allow users to login using usernames and passwords."

**R24.** "The system must allow registered users to reset passwords in case they have lost or forgotten them." **[5]**

**R25.** "The system must allow logged in users to search for academic content using various criteria." **[Pending]**

**R26.** "The system must allow logged in users to watch academic videos." **[5]**

**R27.** "The system must allow logged in users to read academic documents." [5]

**R28.** "The system must allow logged in users to do academic quizzes." [5]

**R29.** "The system must allow students to ask expert consultants and other students questions pertaining to the work in discussion boards after each lesson." [5]

**R30.** "The system must allow students to comment on video lessons to ask questions and answer questions on the lesson." [3]

**R32.** "The system must allow students to redo quizzes." [2]

To be amended: the first try should be the one taken the measure performance levels.

**R33.** "The system must allow students to view their academic progress." [3]

**R34.** "The system must keep a record of the recent activities and resource accesses of its students." [3]

### *Marketing*

**R35.** "The system must allow admin to add and remove sponsor advertisements on the login page of the website." [4] {job advertisements, in house events, external events, sponsor advertisements}

This is not an ad system. These "advertisements" are not systematically rotated. User data is not used to enhance the advertising (make them relate to the user preferences). These are implicitly stored on the pages themselves instead of in a database. It would probably be useless to manage additional overhead to manage a few images in a database. This is almost a gallery in fact, with manage being able to be removed or added.

**R36.** "The system must allow admin to add and remove the logos of the sponsors on the about page of the website." [2]

This is almost a gallery in fact, with manage being able to be removed or added.

### *Gamification*

**R41.** "The system must be able to display willing students on a leaderboard according to performance data." [2]

**R41.1** "The system must provide the user with a choice to appear on the leaderboard."

**R41.1.1** "The system must obtain leaderboard appearance settings from a database."

**R41.1.2** "The system must allow the user to enable leaderboard appearance in their profile."

**R41.1.3** "The system must allow the user to disable leaderboard appearance in their profile."

**R41.1.4** "The system must save leaderboard appearance settings in a database."

**R41.2** "The system must provide a leaderboard ranking students."

**R41.2.1** "The system must rank students according to each subject and grade."

**R41.2.1.1** "The system must rank students according to performance levels."

**R41.2.1.2** "The system must rank students according to provinces."

**R41.2.1.3** "The system must rank students according to difficulty levels."

**R41.2.1.4** "The system must display user ranking as including their province, performance level, and difficulty level."

**R41.2.1.5** "The system must obtain leaderboard data from a database."

This could be done periodically to lessen the impact on database queries, so displayed leaderboards may be rendered at once and stored elsewhere.

Instead of refreshing leaderboard all the time, show date and time of the leaderboard "picture." So the users just refresh the page to get a new one if small increments to the database are made instead of large ones periodically.

**R41.2.2** "The system must order the leaderboard according to performance levels and difficulty level."

**R41.2.3** "The system must display the ranking leaderboard tables for each subject."

**R42.** "The system must be able to reward users with badges as they progress through the content in different levels according their academic progress." **[1]**

To be amended: High progress can unlock a mathematician badge, or high performance can unlock badges modelled after great scientists of past and current ages.

In Moodle, badges are awarded upon course or activity completion. Progress levels are functionality that we would have to implement ourselves it seems. Progress levels can be determined by the number of

activities that are done. The level then would be high if they are close to the course completion. The level affects their progress, i.e. beginners badges (little Einstein), advanced badges (big Einstein)

The intention was to have non-linear behaviour also be rewarded by badges, like fasted test taker, if focused on mathematics then math related. But this could be added later as additional functionality.

{We can operate using the Mozilla Open Badges standard}

**R42.1** “The system must be able to reward users with badges as they finish certain activities.”

These activities must be the most difficult or notable by some arbitrary factor or perhaps set by the teacher as is the case with the way badges are implemented in Moodle. Badges should be mapped out and the milestones needed to achieve them as well (trees and maps).

**R42.1.1** “The system must be able to track user quiz completion in order to award a badge for its completion.”

This should only happen if that quiz has a badge allocated to it. Could badge be recycled amongst other course, as in those that are not explicitly related to the courses by their symbolism or description, as in generic and recyclable course badges. Also the quizzes that have badges could be selected randomly or the setting chosen when the quiz is created. If randomly, the admin won't have to do it.

**R42.1.1.1** “The system must be able to save quiz completion to a database.”

**R42.1.1.2** “The system must be able to check to award a user with a badge after quiz completion.”

**R42.1.2** “The system must be able to retrieve a badge from a database in order to award for a quiz completion.”

The logic used here should include, by random selection perhaps or periodical selection, the possible selection of a course specific badge or a generic recyclable badge. Course specific badges should be left for later iterations of the system. These badges can be made and added to the database at any time, since the logic to retrieve them every now and then would be already implemented (the increases would increase the pool of badges).

**R42.1.3** “The system must be able notify the user that the badge has been awarded to the user for a quiz completion.”

Moodle has the functionality to send notifications about badge rewards through email to the users.

**R42.1.4** “The system must be able to display the badge that has been awarded to the user for a quiz completion.”

This can be done through Bootstrap's modal windows, with some animation added to it, like fireworks. An example could be how random awards are shown in Subway Surfer.

**R42.2** "The system must be able to reward users who have reached certain experience point milestones with badges."

These are like the "hours" put into learning which are rewarded, much like the reward system used Google Play Games' profiles e.g. an amateur award for 16 hours etc. Milestones should be set and mapped out.

**R42.2.1** "The system must be able to access user experience points in order to reward users with badges."

Store the experience points in a session variable once the user has logged in because could be doing numerous activities during their session. Then update these when you access leaderboards? Small incremental database entries or large periodical ones, which is better for performance?

Instead of refreshing leaderboard all the time, show date and time of the leaderboard "picture."

**R42.2.1.1** "The system must be able to obtain experience points from a database in order to award badges."

**R42.2.1.2** "The system must use obtained experience points to award badges to users."

**R42.2.2** "The system must be able to retrieve a badge from a database in order to reward users with badges for experience points gained."

These badges should be different from the generic recyclable badges used for different courses and from the course specific badges. These ones should also be able to be added to the database to increase the pool of badges that can be chosen. These should include a set of badges within each experience point range or one for each milestone or point in the experience point line.

**R42.2.3** "The system must be able notify the user that the badge has been awarded to the user for an experience point milestone."

These milestones should be mapped out e.g. 5 hours gets you this, 15 hours gets you that.

**R42.2.4** "The system must be able to display the badge that has been awarded to the user for an experience point milestone."

**R43.** "The system must be able to give students experience points based on the amount of active hours they put in learning and their progress based on quiz and assessment performance." [2]

Level up! Gamification experience points plugins are available from the Moodle site. The features for this plugin include<sup>[14]</sup>:

- Automatically attributes points to students for their actions
- Block that displays current level and progress towards next level
- Report for teachers to get an overview of their students' levels
- Notifications to congratulate students as they level up
- A leaderboard to display the ranking of the students
- Ability to customise the number of levels, the points they require and their appearance
- Total control over the points earned per action
- Unlock content when a certain level is reached
- Experience points earned per course, or for the entire site
- Support for shortcodes to include in content

**R43.1** “The system must be able to give users experience points after watching a topic video.”

**R43.2** “The system must be able to give users experience points after completing a quiz.”

**R43.3** “The system must be able to give users experience points after reading slides.”

**R43.3.1** “The system must be able to check amount of time spent reading slides.”

**R43.3.2** “The system must be able to check if the user got to the end of the slides.”

**R43.3.3** “The system must be able to see the amount of time spent on each page of the slides.”

**R43.3.4** “The system must allocate experience points based on time spent on the slides.”

**R43.3.5** “The system must show experience points are given for reading slides.”

**R43.4** “The system must show that the user has been given experience points for slides, video, and quiz activities.”

**R44.** “The system must be able to allow students to compete with each other in individual or group tournaments, by performing designed assessments in a specified period of time.”

**[Pending]**

**R44.1** “The system must allow users to join teams in order to compete in group assignments.”

**R44.2** “The system must allow teams to participate together in completing group assignments.”

**R44.3** “The system must allow winning teams to progress to the next group assignments in the tournament.”

**R44.4** “The system must allow losing teams to be disqualified from competing in the next group assignments of the tournament.”

**R44.5** “The system must disqualify a team if a limit of members is not reached online when participating in a group assignment.”

**R44.6** “The system must notify the team if they have been disqualified from the tournament.”

**R44.7** “The system must allow the best team to win the tournament and receive a prize.”

**R44.8** “The system must notify the best team that they have received a prize.”

**R45.** “The system must gradually increase the difficulty of the quizzes and assessments based on the student's increasing performance levels (a difficulty level system).” **[Pending]**

To be amended: Quizzes will start getting boring if a student has hit their assessment ceiling. Increasing this ceiling gradually may keep the interest of performing students (a level system). Timing the assessment activities can add difficulty as well. The system must allow questions to be categorized according to difficulty levels and these should be made beforehand (more than enough of them) so that they can be chosen dynamically. Increasing questions and scoring low difficulty questions with lower marks could also be a solution.

But students won't be doing the same thing over and over though? Perhaps...revision? Making quizzes they haven't done before more difficult might make it a little cumbersome, no?  
How are performance levels measured? According to experience points? Or course activity as well...

Question difficulty levels can also be indicated by that learning standard thing in CAPS

**R45.1** “The system must monitor the student's performance levels.”

**R45.1.1** “The system must consider the student's experience points in measuring their performance levels.”

**R45.1.2** “The system must consider the previous scores the student has achieved in the course's quizzes.”

**R45.2** “The system must adjust the quiz difficulty according to the performance levels of the students.”

**R45.2.1** "The system must choose/add more difficult questions for the quizzes if the performance levels of the student are high enough."

**R45.2.2** "The system must adjust the scores of questions of varying difficulties."

**R45.2.2.1** "The system must have lower difficulty questions be scored with lower points."

**R45.2.2.2** "The system must have high difficulty questions be scored with higher points."

**R45.3** "The system must display the difficulty levels of the questions next to the question."

**R46.** "The system must be able to allow students to share their performance data on social media to their friends and family." **[Pending]**

**R47.** "The system must be able to allow students to change their avatars through the badges they have earned or custom avatars." **[Pending]**

Possible copyright issue: Khan Academy idea?

**R48.** "The system must allow users to store all their badges and awards in an accessible inventory." **[1]**

This can be done using Moodle's own backpack setup in which badges shared to the OpenBadges (<https://backpack.openbadges.org/backpack/welcome>) site are placed into the user's backpack.<sup>[13]</sup>

**R49.** "The system must make special items attainable after completing certain assessments or tournaments (either individual or group)." **[Pending]**

These special items can be displayed in team pages.

**R50.** "The system should allow students to form their own teams for tournaments." **[Pending]**

**R50.1** "The system should allow students to become team leaders of their teams."

**R51.** "The system must allow experience points being able to be exchanged for hints to certain hintable questions." **[Pending]**

**R52.** "The system must have a personal progress leaderboard in the user's profile to show how well you have been doing in quizzes or assessments of particular subjects - like the games in Lumosity." **[Pending]**

**R53.** "The system must hold invitations to random but relevant challenges which end after a certain period of time." **[Pending]**



- Reward suggestions:

{"Scientist collector cards for science (Elon Musk rare card), Business people for business studies (Bill Gates rare card), Marvels of nature of tourism, Asteroids that can be minded with some other tool that can be earned or exchanged in a card exchange"}

{Emoticons/emojis like badges for some behavioural badges e.g. most replies in the discussion board could give you a 'cool guy emoji', an owl for those that have tendencies to complete activities and learn at night.}

**R54.** "The system should ascertain how stressed or overworked the student is with a set of questions after logging in, and set difficulty and activity duration levels accordingly." **[Pending]**

To be amended: Asking how many hours the users have slept to gauge how much activity and focus could be expected, and change the frequency of wakeup or 'are you still with us' calls from the chatbot, and also to adjust the length of quizzes and perhaps the complexity of the questions. (Question complexity can be determined using those CAPS guidelines about question difficulty --> from high school, do you remember them?)

These should be taken into account in the leaderboards to prevent students from cheating trying to reducing difficulty levels of the questions. The leaderboards should place students with the highest experience points and highest question difficulties.

### *Chatbot*

**R55.** "The system must detect user inactivity during assessments and query them to ascertain if they need help, or in the case of hinted questions, a hint." **[3]**

To be amended: "Are you still there, Dingan?"

**R55.1** "The system must keep track of long pauses and lack of input."

This can be done through Javascript by using Web workers. A web worker runs in the background on the page, without affecting the performance of the page<sup>[12]</sup>. We can use the web workers to time periods between user input and after reaching a certain threshold, deploy the chatbot.

**R55.2** "The system must keep track of the current user activity to determine the appropriate query."

When a certain page is loaded, information about the page type i.e. home or FAQs, is sent to a web app in the background, which would then assist the user appropriately in the case when the user is inactive or needs help. The information is used to assess a data structure of appropriate actions.

**R55.2.1** "The system must keep track of the current user page to determine the appropriate query."

**R55.2.2** "The system must keep track of the current user quiz to determine the appropriate query."

**R55.3** "The system must give users hints when stuck on a question."

**R55.3.1** "The system must allow for the user to click a hint button, enabled according to the availability of the hint itself."

**R55.3.2** "The system must ask the user if they require a hint."

This after asking if the user is there.

**R55.3.3** "The system must show the user the hint using the chatbot interface."

**R56.** "The system must provide the user with the ability to query about anything they need from the website using their own sentences." **[1]**

This serves as an interactive guide for the users. This functionality could not be restricted to page related queries, but could eventually reach out to the whole site and become integrated with a search function.

**R56.1** "The system must allow users to input queries into the chatbot interface."

**R56.2** "The system must be able to analyse queries entered in order to produce correct response."

This is where a Natural Language processing API comes in handy.

**R56.3** "The system must be able to display output to users to satisfy queries."

If unable to satisfy queries, the chatbot should let the user know.

**R57.** "The system must introduce the website to new users in a conversational manner, so as not to bombard them with loads of information and unimportant details." **[2]**

To be amended: This can be done using Landbot.io ([www.landbot.io](http://www.landbot.io)); a demonstration of this is here: <https://www.youtube.com/watch?v=DRgLDT0-en4> . The service is free to use, has a drag and drop interface and a number of other options to explore from (cite: [www.landbot.io](http://www.landbot.io) ; how do you cite a chatbot?). However, an alternative might need to be found as it is only free for 100 chats/per month and 30 messages per bot.

**R57.1** "The system must be able to highlight parts of the page to indicate focus."

**R57.2** "The system must be able to display information using the chatbot interface popups."

This could be done using modals Bootstrap modals.

**R58.** "The system must be able to notify users who have switched tabs to get back to their activities." [1]

To be amended: This can be done using browser notifications.

**R58.1** "The system must keep track of the amount of time spent offsite after switching sites."

**R58.2** "The system notify the user to get back to work using browser notifications."

**R59.** "The system must enable users register in a conversational manner, as an alternative to forms." [3]

To be amended: The children might like this feature, instead of filling out a form declaring your subjects and interests, a chatbot meets them on the registration page and asks them, in a conversational manner e.g. "Hi Tracy, I'll be helping you with the registration process. Which grade you in?"

This can be done using Landbot.io ([www.landbot.io](http://www.landbot.io)). Using this, the background can be a video of computer lab session or children/people interacting with computers in an academic session (cite: <https://app.landbot.io/app/bot/52577?view=design&tab=colors>). Or an image.

**R59.1** "The system must capture form information using the chatbot interface."

**R59.1.1** "The system must obtain target form questions for the chatbot to obtain user information."

**R59.1.2** "The system must ask for user information using the chatbot interface."

The information requested should be information required by the classical form.

**R59.1.3** "The system must allow user information to be entered by users."

**R59.1.4** "The system must validate captured user information and ask for corrections."

**R59.2** "The system must enable the user to toggle off the chatbot interface."

**R59.2.1** "The system must provide a toggle switch to toggle the chatbot interface."

**R59.2.2** "The system must provide a registration form for the user to provide input."

**R59.3** "The system must store user information into the database."

**R60.** "The system must ask for user feedback in a conversational manner." [1]

**R60.1** "The system must allow the user to enter user feedback using the chatbot interface."

**R60.2** "The system must store the user feedback in a database."

**R60.3** "The system must create a user feedback report to send to the admin."

**R60.3.1** "The system must generate a feedback report."

Using the database delegated for this. This should be scheduled.

**R60.3.2** "The system must send the report to the administrator."

**R61.** "The system must replace FAQ lists with interactive conversation to promote user engagement." **[Pending]**

**R62.** "The system must wish users a happy birthday through the chatbot." **[1]**

To be amended: Example: "Hi Mpho, a little bird told me it was your birthday today. I'd like to wish you a happy 15th <cake emojis." This can be done by obtaining date of birth information from the database.

**R62.1** "The system must be able to retrieve birthday information from the database."

This information can be saved in the database and retrieved there. This is an optional nicity.

**R62.2** "The system must display a birthday wish using the chatbot interface."

**R62.3** "The system must notify teammates of the user's birthday." **[Pending]**

**R63.** "The system must welcome back registered users to the site and ask them if they want to go back to their last activity." **[Pending]**

**R64.** "The system must respond to students to notify them if an expert consultant is not available through the chatbot in the discussion board when a message is sent to them. {this is like what the slackbot does}." **[Pending]**

## **Pending**

Pending requirements will just have to be accommodated between the iterations during the "accommodating requirements change" activity. Changes to the requirements could possibly result to some redesign of the system architecture and use cases allocated to each iteration. Keep in mind the priorities of those requirements.

## **Resources**

- [1] Moodle.com. (2015). *Performance and Scalability* [Online]. Available at: [https://docs.moodle.org/dev/Performance\\_and\\_scalability](https://docs.moodle.org/dev/Performance_and_scalability) (Accessed: 21 May 2018)
- [2] Moodle.com. (2013). *Test site generator* [Online]. Available at: [https://docs.moodle.org/dev/Test\\_site\\_generator](https://docs.moodle.org/dev/Test_site_generator) (Accessed: 21 May 2018)
- [3] Moodle.com. (2013). *Test course generator* [Online]. Available at: [https://docs.moodle.org/dev/Test\\_course\\_generator](https://docs.moodle.org/dev/Test_course_generator) (Accessed: 21 May 2018)
- [4] jmeter.apache.org. (2018). *Apache JMeter* [Online]. Available at: <http://jmeter.apache.org> (Accessed: 21 May 2018)
- [5] Stallings, W. (2015). *Operating Systems Internals and Design Principles*. 8th Edition. New Jersey: Pearson Education, page 73-74
- [6] Saenz, J.G. (2015). *New (and free) uptime monitoring tool*. [Online]. Available at: <https://moodle.org/mod/forum/discuss.php?d=324529> (Accessed: 6 June 2018)
- [7] toodle.org. (2017). *Monitoring*. [Online]. Available at: <http://toodle.org/> (Accessed: 6 June 2018)
- [8] Moodle.com. (2015). *Security: Brute-forcing login*. [Online]. Available at: [https://docs.moodle.org/dev/Security:Brute-forcing\\_login](https://docs.moodle.org/dev/Security:Brute-forcing_login) (Accessed: 6 June 2018)
- [9] Moodle.com. (2016). *Transitioning to HTTPS*. [Online]. Available at: [https://docs.moodle.org/35/en/Transitioning\\_to\\_HTTPS](https://docs.moodle.org/35/en/Transitioning_to_HTTPS) (Accessed: 6 June 2018)
- [10] Stallings, W. (2015). *Operating Systems Internals and Design Principles*. 8th Edition. New Jersey: Pearson Education, page 82
- [11] Moodle.com. (2012). *Anti-virus*. [Online]. Available at: <https://docs.moodle.org/23/en/Anti-virus> (Accessed: 26 June 2018)
- [12] w3schools.com. *HTML5 Web Workers*. [Online]. Available at: [https://www.w3schools.com/html/html5\\_webworkers.asp](https://www.w3schools.com/html/html5_webworkers.asp) (Accessed: 1 July 2018)
- [13] Moodle.com. (2015). *Badges FAQ*. [Online]. Available at: [https://docs.moodle.org/28/en/Badges\\_FAQ#What.27s\\_the\\_difference\\_between\\_a\\_backpack\\_and\\_a\\_collection.3F](https://docs.moodle.org/28/en/Badges_FAQ#What.27s_the_difference_between_a_backpack_and_a_collection.3F) (Accessed: 3 July 2018)
- [14] Frederick Massart. *Blocks: Level up!*. [Online]. Available at: [https://moodle.org/plugins/block\\_xp](https://moodle.org/plugins/block_xp) (Accessed: 3 July 2018)