Thomas Moseley, Kassandra Coan, Adam Gorman, Shawn Williams

Team Ludus

Abstract

There is no reason that learning cannot be fun and engaging. In today’s game-saturated world, it’s time for a learning technology that makes learning at a collegiate level enjoyable. This is how the ball gets rolling.

Ludus

Software Design Document, version 1.0

Table of Contents

[Revision History 3](#_Toc342598767)

[1 Introduction 4](#_Toc342598768)

[1.1 About Ludus 4](#_Toc342598769)

[1.2 Purpose 4](#_Toc342598770)

[1.3 Document Conventions 4](#_Toc342598771)

[1.4 Intended Audience and Reading Suggestions 4](#_Toc342598772)

[1.5 Product Scope 5](#_Toc342598773)

[1.6 References 5](#_Toc342598774)

[1.7 Video Presentation 5](#_Toc342598775)

[2 Overall Description 7](#_Toc342598776)

[2.1 Product Perspective 7](#_Toc342598777)

[2.2 Product Functions 7](#_Toc342598778)

[2.3 User Classes and Characteristics 8](#_Toc342598779)

[2.4 Operating Environment 10](#_Toc342598780)

[2.5 Design and Implementation Constraints 10](#_Toc342598781)

[2.6 User Documentation 10](#_Toc342598782)

[2.7 Assumptions and Dependencies 10](#_Toc342598783)

[3 External Interface Requirements 11](#_Toc342598784)

[3.1 User Interfaces 11](#_Toc342598785)

[3.2 Hardware Interfaces 12](#_Toc342598786)

[3.3 Software Interfaces 13](#_Toc342598787)

[3.4 Communications Interfaces 13](#_Toc342598788)

[4 System Features 14](#_Toc342598789)

[4.1 Scheduling 14](#_Toc342598790)

[4.1.1 Description 14](#_Toc342598791)

[4.1.2 Priority 14](#_Toc342598792)

[4.1.3 Functional Requirements 14](#_Toc342598793)

[4.2 The Reward System 14](#_Toc342598794)

[4.2.1 Description 14](#_Toc342598795)

[4.2.2 Priority 15](#_Toc342598796)

[4.2.3 Functional Requirements 15](#_Toc342598797)

[4.3 Communication Forums 16](#_Toc342598798)

[4.3.1 Description 16](#_Toc342598799)

[4.3.2 Priority 16](#_Toc342598800)

[4.3.3 Functional Requirements 16](#_Toc342598801)

[4.4 Course Management 17](#_Toc342598802)

[4.4.1 Description and Priority 17](#_Toc342598803)

[4.4.2 Priority 17](#_Toc342598804)

[4.4.3 Functional Requirements 17](#_Toc342598805)

[5. Other Nonfunctional Requirements 19](#_Toc342598806)

[5.1 Performance Requirements 19](#_Toc342598807)

[5.2 Safety Requirements 19](#_Toc342598808)

[5.3 Security Requirements 19](#_Toc342598809)

[5.4 Software Quality Attributes 19](#_Toc342598810)

[5.5 Business Rules 19](#_Toc342598811)

[6. Other Requirements 20](#_Toc342598812)

[Appendix A: Glossary 20](#_Toc342598813)

[Appendix B: Use-Case Diagram 21](#_Toc342598814)

[Appendix C: Class Diagram 22](#_Toc342598815)

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| Initial Draft | April 10, 2012 |  | 0.1 |
| Final Draft | December 6, 2012 |  | 1.0 |

# 1 Introduction

## 1.1 System Overview

Ludus: Noun (Latin)

1. School

2. Game, Sport, Play

3. Fun

The name Ludus was inspired by the book Ready Player One, written by Ernest Cline, published by Random House in August, 2011. In this book, most interaction in the world takes place in an online simulation called the OASIS. The protagonist, Wade Watts, is a student on the virtual world Ludus, a world populated only with schools. The book turns on Wade’s realization that the word Ludus not only means School, but means Sport and Game as well.

For the most part, the students of today grew up with games everywhere. From different gaming platforms, such as Xbox, PlayStation, and the Wii, to mobile device games, like Angry Birds. The feedback loop of playing for reward is a strong one, ingrained in the students of today.

With Ludus, we’re adapting the school world to the real game world students are exposed to from childhood. Just as Mario has his power-ups, so do students, for achievements that their teachers or fellow students create. Students understand these goals, and they have fun working toward them.

There’s no reason learning cannot be fun, is there?

## 1.2 Supporting Materials

Ludus System Requirements Specification. <http://ludus.azurewebsites.net/Documents/Ludus%20SRS.docx>

## 

## 1.3 Document Conventions

Throughout this document, higher-level requirements are assumed to be decomposed into detailed requirements before development.

Throughout this document, the feminine pronoun (she, her, or hers) will be used to indicate a person or action. This implies no bias toward either sex, but is used for ease of writing and reading, over masculine (he, him or his) or neutral (their or theirs).

Through this document, the term “activity” and “activities” shall be used to indicate classroom-based activities, such as a lecture, homework, quiz, examination, or project. Individual use of these terms indicates that the narrative applies only to those types noted, and not to the other types.

This document shall use the name Ludus, and the term “system” interchangeably.

## 1.4 Intended Audience and Reading Suggestions

This document is intended to be used for guiding the development of the gamified online classroom experience. There are general descriptions that will be useful for an entire team’s understanding, from the management team through the development team.

Upon completion of reading this document, one will have a full understanding of the goals and features the completed product should include.

One should read the document from start to finish. The material is set up so that the general product description is given, then the details of how the product is presented (i.e. the interface), followed by nonfunctional requirements, that may be of particular use to the developer when deciding how much time should be devoted to code optimization.

## 1.5 Product Scope

Ludus will provide an intuitive and fun interface for students and staff to coordinate their classroom requirements. Centering on a session calendar, Ludus treats schoolwork the way students do, as a series of tasks and deadlines.

On top of this, Ludus adds a layer of “gamification”, defined as “the concept of applying game-design thinking to non-game applications to make them more fun and engaging.” The goal of Ludus is to provide an interface where a school, department, instructor, or student can create game-style challenges, where completion of these goals can result in rewards.

The combination of these two concepts, of addressing school as a series of date-driven challenges, while adding the gamification experience, will engage students at all levels, enhancing their learning experience.

## 1.6 References

<http://gamification.org/wiki/Gamification_of_Education> - A link to gamification.org, a site dedicated to information on the gamification of non-gaming systems.

[www.openid.net](http://www.openid.net) - A foundation advocating an open authentication system. The OpenId standard will be our standard authentication interface.

<https://en.gravatar.com> - A system for designating a “global avatar”, associated with an email address.

[https://bitbucket.org](https://bitbucket.org/)- A project management host/repository, used for development requirements management and to host the code repository.

<http://twitter.github.com/bootstrap/> - A set of user interface components that give a uniform and modern look and feel.

<http://stackexchange.com> – A set of message forums, that apply the concepts of gamification to peer-to-peer communications.

## 1.7 Video Presentation

A video presentation on Ludus can be found at <http://www.youtube.com/watch?v=XzrVkFidff8>.

# 2 Overall Description

## 2.1 Product Perspective

Ludus is a stand-alone classroom management system. Currently, different classroom management systems exist, such as Blackboard and Angel Learning. These systems are course-centric, not student/schedule centric. That is, students navigate between courses to attempt to determine requirements and deadlines. This results in students missing assignments or being frustrated with a non-responsive product.

The functionality that makes Ludus unique is the element of gamification. Instead of the classroom experience being a cold series of assignments, projects, and examinations, GC allows users at different levels to complete challenges for badges, which represent an internal currency for students, used to acquire rewards.

Upon entry to Ludus, a student will be presented with the home-page view. This view will contain the following information:

1. A calendar displaying the [activities](#id.ks7rssuxs1t8) to be worked over an upcoming period of time, either the day, week, or month.
2. A list of their current game goals. This will be a list of badges available, as well as their current balance of reward bucks.
3. A list of new goals, filterable by the goal source (institution, faculty, students).
4. Links to class resources.
5. A menu system for gaining more information.

As activities come into scope, that is, where the student decides to take notice, the student can customize the activity with an estimate of the amount of time they expect to devote to it, over what period of time. This allows the student to budget their time and reduce the risk that deadlines will come up without addressing an activity.

Through gamification, individual activities or goals can be assigned badges. For example, a student might get a “badge” for turning in an assignment before a previous assignment’s due date, or a badge for completing every assignment early. Students can also use the reward bucks for badges earned to challenge fellow students to stretch themselves, so a student could challenge another student to get a grade better than 90% than, for example, a study-group partner.

Badges earned will be visible throughout the system, and some will have limited audience, that is one for an entire semester or for a class, such as a “highest grade” badge for a class, where a student gets the badge after the first assignment and it moves to other students as they take over this position.

Reward Bucks can be spent at a rewards page, in exchange for different rewards and prizes.

The net effect is that through an easy user-interface for managing goals and a gamified system for engaging students, Ludus will increase student involvement in the education process, and give faculty some way to incentivize students, short of grades.

## 2.2 Product Functions

As mentioned in section 2.1, Ludus is a web-based learning system, and will be available to all students and faculty at an institution at once. Some functions that Ludus will provide include:

* The Calendar View, where students can see their upcoming assignments and events.
* An integrated time estimation and logging system, so that students can, if they choose, work out schedules for getting their school (and other) work done.
* A system of interactive forums, where authorized individuals can discuss various subjects.
* A “Badge” based reward system, where individuals can create goals, which reward achievement.
* A reward store, so that students can cash in their reward points for various prizes, such as school-based gear and trophies.
* Social media functionality, such as user pages, where students can show their achievements and follow other students, as well as online text-chatting and messaging functionality.
* Assignment management, where students can complete assignments and submit them for evaluation by faculty.
* Course-management tools, so that faculty can develop and assign homework, reading and/or viewing material, and perform assessments of each student’s comprehension.
* Online testing, where combinations of test questions, answer options can be submitted, so that students can take these tests and get rewarded badges for performance.

## 2.3 User Classes and Characteristics

Access to Ludus is limited to those with valid credentials. Ludus does not support anonymous access. There are four roles defined in Ludus:

* Administrator
  + Perform maintenance tasks
  + Simulate being other roles
  + Configure terms and courses
  + Configure institutional branding and standards
  + Designate faculty members
  + Designate Students per session
  + Approve account access for registering students to their institution.
* Faculty
  + Create sections for courses for a particular term
  + Designate certain students as aides, granting them faculty permissions for a particular class or section.
  + Create and manage coursework
    - Syllabi
    - Activities
    - Videos
  + Create grading schedules, based on weighting of course elements
  + Create and manage course/section forums, graded and ungraded, where students can interact and express opinions.
  + Monitor and hide inappropriate input.
  + Given a reward point budget for each section, create course badges
  + Create rewards available only to students in a class
  + Allocate extra credit based on badge rewards
  + View the backlog of students’ coursework, to allow for adjustment of schedules and reevaluation of deadlines
* Students
  + Manage their course schedules through the Calendar interface
    - Designate internal goals for progress
    - Estimate time to completion for each item
    - Mark time spent for each item, shown against the estimate to allow for adjustment to estimation
    - Prioritize items, to allow for handling of conflicting deadlines
    - Submit items for evaluation, marking them as complete and removing them from the work backlog
  + Contribute material to course/section forums
    - Format text using a rich HTML editor interface
    - Link to external documents or systems
    - Vote on forum posts, either positively or negatively. Positive votes may apply to the earning of badges.
    - A forum thread can be started as “Asking a Question.”
      * On such a forum thread, the asking student may mark a single response as “Answering the Question.” This action rewards points to the asker and first accepted answerer.
  + Create badges for rewards, using their own budget of earned points as the payment.

## 2.4 Operating Environment

Ludus will operate on a Windows Server 2008 (or later) server instance, in a non-load-balanced mode. Ludus will use a SQL Server 2008 (latest SP) database server.

## 2.5 Design and Implementation Constraints

* The system shall comply with Section 508 of the United States Workforce Rehabilitation Act of 1976.
* The system shall collect no personally identifying information (PII).
* The system shall use a SQL Server 2008 (Current SP) database.
* For non-mobile consumers, the system shall support monitors with a resolution width of no less than 1024 pixels.
* The system shall use Secure Socket Layer (SSL) communication to ensure that data is transmitted to the system in a secure fashion.
* The system shall be constructed using an object oriented programming (OOP) language, such as C#, following industry-standard naming and programming conventions.
* Version 1 will be available to run in a hosted mode only. Future versions will include a purchased local hosting solution.
* Users will be required to have an OpenId login. OpenId is an open source login system. OpenId providers include Google, Yahoo, Blogger, and Verisign. For more information on OpenId, refer to <http://openid.net/>.
* Users’ personal “Avatars” will be acquired using Gravatar. For more information on Gravatar, refer to <http://en.gravatar.com>.

## 2.6 User Documentation

Ludus will be delivered including standard web-based documentation, including hovering, context-sensitive tooltips as well as help links delivering written help content. A list of frequently asked questions (FAQ) will be provided on initial delivery. This list will be expended in the future, through the placement of curator-edited content in a fixed location.

Users will have the option to be guided through a tour upon their first visit.

Ludus shall incorporate an announcements page, where the system administrator, an institution administrator, or a faculty member can post announcements for viewing on an individual’s next login. This can be used for outage notifications or for “what’s new” information on product updates.

Configuration documentation will be provided separately for Institutional Administrators.

## 2.7 Assumptions and Dependencies

These listed assumptions are provided to outline factors as yet unknown which could affect the construction or delivery of Ludus.

* The user of Ludus will be using a browser compatible with HTML5 and CSS 3.0. An active internet connection is required to use Ludus.
* For version 1.0, no special consideration shall be given to mobile platforms.

# 3 External Interface Requirements

## 3.1 User Interfaces

* Login Page
  + Allows a user to log in with OpenId credentials. This page will incorporate links to various OpenId login servers.
  + Provides a “register” link, to allow a user to log in for the first time. This registration will allow for setting up OpenId credentials and requesting they be linked to an institution’s student id.
* Home Page
  + Presented to Faculty and Students upon logging into the system.
  + Consists of the following User Interface elements
    - Calendar View
      * Provides Views for different time spans
        + Day
        + Week
        + Month
        + Semester
      * Remembers the last time span the user had visible
      * Scales data display and data for each day
        + For smaller day views, summary data is shown, as well as a link to a day view.
      * Time budgeting
        + When the user clicks on a task/goal/deadline, they can specify the amount of time they want to budget for that activity
      * Time logging
        + When the user clicks on a task/goal/deadline, they can specify the amount of time they have spent on that activity
    - Social Panel
      * Shows a chat status window, including the ability to initiate chats with individuals or groups.
      * Shows favorite forum channel traffic
    - Goals At a Glance
      * Allows the user to see a list of their upcoming deadlines and goals, as well as their progress.
* Course Page
  + The focal page for course information. The page is split into two panels.
    - Course Outline
      * Presents the course in two views
        + A calendar view, where all activities for the course are present

Can be filtered to show a subset of activities, by activity type, completion status.

* + - * + A list view, displaying a list of all course activities

Can be filtered to show a subset of activities, by activity type, completion status.

Can be sorted by activity type, completion status, etc.

* + - * For students, information such as grades, progress toward goals, and faculty notes will be added to displays
      * For faculty, the following additional functionality is provided
        + The ability to add activities is provided, with the appropriate activity editors.

For quizzes and examinations, the ability to specify a pool of questions, with answers, and select how many are used.

For lectures, faculty can upload material or provide links to outside systems, such as YouTube.

Any graded activities (quizzes, examinations, homework, and projects) are subject to badges and rewards.

* + - * For all users, the ability will be provided to add their own personal items.
    - Item detail - Shows the detail for the currently selected item.
* User profile page
  + Enrolled classes
  + Badges/Achievements
  + Reward Points
  + Rewards claimed
  + Rewards posted
    - Include a link to post a new reward
  + A link to the Gravatar image linked to the user’s ID. (see [www.gravatar.com](http://www.gravatar.com))

## 3.2 Hardware Interfaces

No specialized hardware interfaces are anticipated.

## 3.3 Software Interfaces

Ludus shall interface with different authentication systems implementing the OpenId standard. The system shall establish linkages between OpenId credentials and institution identities.

Ludus shall interface with Gravatar to present avatars for users based on their email address.

## 3.4 Communications Interfaces

Standard web communication tools:

* In site notifications
  + Version 1.0 will display a user’s school identification in the header to allow the student to verify that she is logged into the correct system.
  + Upon account creation, students will receive an email at their registered school email address, verifying login information.
* HTTPS
  + Ludus shall use Secure Socket Layer encryption for browser communication.
* We will use delegated login for users, so they can sign in via OpenId.

# 4 System Features

System Features, described in this section, will be prioritized using a MoSCoW (Must have, Should Have, Could have, or Want), both for version 1 and the final product, as well as by the relative level of effort to implement each feature.

## 4.1 Scheduling

### 4.1.1 Description

The primary function of Ludus is to operate as a classroom management system; regardless of the fun of the system, if students cannot manage their schedules, the system is of limited use.

### 4.1.2 Priority

Must have, Version 1

### 4.1.3 Functional Requirements

1. All users shall have access to a schedule view.
2. The schedule view shall provide different time spans for viewing, limited to Day, Week, Month, and Term.
3. The schedule system shall provide a calendar view, with an appropriate number of items shown for each day, and a button provided to display more items.
4. The schedule system shall provide a list view, with all items for a particular time span shown.
5. The schedule list view will be sortable.
6. The schedule view will allow the user to filter items based upon various criteria.
7. The schedule system will allow a user to whom an activity is assigned to estimate the amount of time budgeted for that activity.
8. The schedule system will allow a user to whom an activity is assigned to record the amount of time spent for that activity.
9. The schedule system shall allow a user to view the detail on any item visible to them.
10. The schedule system shall allow any user to create personal items in their schedule.
11. The schedule system will allow any user to share personal items with other users within the same institution.

## 4.2 The Reward System

### 4.2.1 Description

Badges are rewards that can be posted and achieved by participants. Badges can be created by anyone with access. Hand-in-hand with Badges are Reward Points, which, among other things, are a translation of an achievement to a tangible or intangible asset. In short, Badges are outward symbols of achievement, visible to peers and faculty, showing one’s progress toward goals and exceptional performance. Reward points are the currency that these achievements yield.

Together, Badges and Reward Points constitute the reward system.

Faculty will have a fixed reward point balance for each class, and can design a reward system accordingly. These can be available to all sections of a particular class, or to individual sections. For instance, an instructor might have a “First Project In” Badge for all of her students, or a “Great Response” badge for a forum post.

Additionally, an institution might have institution-wide rewards, like an “I Have an Opinion” badge for any person who makes more than a set number of posts in forums.

A badge system is a standard methods of gamifying a system. For version 1.0 of Ludus, the Badge system is our primary, but not sole, method of gamification.

### 4.2.2 Priority

Must Have, Version 1

### 4.2.3 Functional Requirements

1. The system shall allow the institution administrator to allocate a budget of reward points to a course based on a per-student formula.
2. The system shall allow for reward creation to be on an institution, course, section, or group organizational basis. Users not in the appropriate organization will not be eligible for the rewards.
3. The system shall allow the designation of single-owner badges inside a single organization.
4. The reward points for a single-owner badge shall not be removed when the award transfers to a different user. The reward points will not be reissued upon regaining of a single-owner badge.
5. The system will allow for the creation of single-earn badges, where each student is limited to a single instance of the badge, but multiple students may possess the badge.
6. The system will allow for the creation of multiple-earn badges, where a student may possess multiples of the same badge.
7. The system will allow for badges to be created based on various criterion, including the earning of additional badges.
8. The student will allow a student to create badges and issue rewards/bounties based to other students based on performance.
9. The system shall display all of a student’s current badges on the student’s profile screen.
10. The system shall allow the student to schedule time toward the achievement of a badge.
11. The system shall allow for students to create their own homework, using flash cards or other student-generated materials, for reward points from the student’s own budget.
12. The system will allow instructors to use course/section badge awards to be used as extra credit.

## 4.3 Communication Forums

### 4.3.1 Description

#### The primary online method for inter-section communication is the message forum. This is not a unique requirement.

The Ludus communication forum system shall draw on the forums of StackExchange for inspiration. In addition to the normal message traffic, individuals will have the opportunity to vote threads/responses up or down. In a forum designated as a question/answer forum, students will be able to vote for answer, and the person originating the thread will be able to select the “best answer.”

### 4.3.2 Priority

Must Have, Version 1

### 4.3.3 Functional Requirements

1. The system shall allow an institutional admin to create one or more institution-wide forums.
2. The system shall allow any user to contribute to an institutional forum.
3. The system shall allow a faculty user to create one or more course/section-wide forums.
4. The system shall limit contributions to a course/section-wide forum to students in the course/section, faculty for the course/section, and institutional administrators.
5. The system shall allow for a forum to be created as either a discussion or question/answer forum at creation time.
6. The system shall not allow for a forum to change types from discussion to question/answer, or vice-versa.
7. The system shall allow a student to flag any forum entry as objectionable.
8. The system shall notify the institution administrator and, if applicable, faculty when a forum entry has been flagged as inappropriate.
9. The system will provide a report to faculty on forum usage for grading purposes.
10. The system shall allow any user with permissions greater than student to remove a post flagged as inappropriate.
11. The system shall automatically isolate (hide) any forum posting with three or more inappropriate reports. This isolation shall include any sub-messages.
12. The system shall provide a web-based editor for forum entries.
13. The system will not allow the addition of file uploads with a forum post.
14. The system will allow html-based links to external systems from forum posts.
15. The system will allow html-based embedded content from external systems into forum posts.
16. The system will allow faculty to create badges for forum activity.
17. The system will allow faculty to assign award points, not badge related, for forum activity.
18. The system will allow faculty to include a reward-point bonus per post.
19. The system will allow faculty to include a reward-point bonus per up-vote, and a reward-point penalty up-vote.
20. On a question/answer forum, the system shall allow students to pose questions.
21. On a question/answer forum, the system shall allow students to post a reward-point “bounty” for a correct answer. This bounty is removed from their account at the time the reward is posted.
22. Bounty questions shall be time-limited. At the expiration of the time limit, 75% of the bounty shall be returned to the student.
23. The student posting a bounty-question shall mark one question as the best answer. The poster of this answer shall receive the bounty.
24. Once a bounty has been rewarded, a change of best answer has no value.

## 4.4 Course Management

### 4.4.1 Description and Priority

While students are why the system exists, faculty needs an interface to do their work that is intuitive and friendly as well.

The goal with course management is it to provide an interface for faculty to create the course framework for their students.

### 4.4.2 Priority

Must Have, Version 1

### 4.4.3 Functional Requirements

1. The system shall allow faculty and institution administrator to remove students from courses, sections, or the institution.
2. The system shall allow an institutional administrator to add courses to a term.
3. The system shall allow an institutional administrator to designate faculty for a course.
4. The system shall allow faculty to create sections for a course.
5. The system shall allow faculty to add students to their courses and sections from the list of student profiles.
6. The system shall allow faculty to create link and document libraries for classes/sections.
7. The system shall allow faculty to create activities for each section
8. The system shall allow faculty to schedule due-dates for activities
9. The system will allow faculty to schedule “become visible” dates for activities, to keep activities hidden until a particular date.
10. The system will allow faculty to create an examination/quiz.
11. The system will allow faculty to create examination/quiz question pools, including true/false, multiple-choice, short answer, and essay questions.
12. The system will present quizzes to students using a subset of questions provided.
13. The system will grade quizzes, with the exception of essay questions, at answer times.
14. The system shall allow faculty to create homework assignments.
15. The system shall require faculty to classify each activity type.
16. The system shall allow faculty to grade individual activities.
17. The system shall apply grade weighting, based on activity type classification and faculty input, to present a running grade for coursework to each student.

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

Average system response for any page load shall not exceed 2 seconds.

When a badge is earned, the student shall be notified on the next page load. If the earning takes place when the student is offline, an alert will appear on their next login.

## 5.2 Safety Requirements

No requirement for safety consideration is anticipated.

5.3 Security Requirements

The system shall not store any personally identifying information. Each student can create an alias for their user id. Any user may report a user id as inappropriate. The user id will be reported to the institution and system administrator for action.

The system shall neither store nor ask for password information.

The system shall use Secure Socket Layer (SSL) communication for all internal and external communications.

## 5.4 Software Quality Attributes

Development of Ludus will be done using industry standard quality measures. Code shall all conform to the best-practices of C# coding standards. External interfaces shall be accomplished using external modules and services, where appropriate. The system shall not directly interface with any other system.

Code formatting shall be accomplished using Cascading Style Sheets (CSS), both to fulfill requirements Section 508 of the United States Workforce Rehabilitation Act of 1976, as well as to provide multi-browser compatibility.

## 5.5 Business Rules

An institution administrator can impersonate a member of the faculty inside Ludus, in case the faculty member is unavailable.

Students may be designated as student aides for a course or section, providing these aides full faculty access for that class/section. They do not function as faculty outside of the course/section.

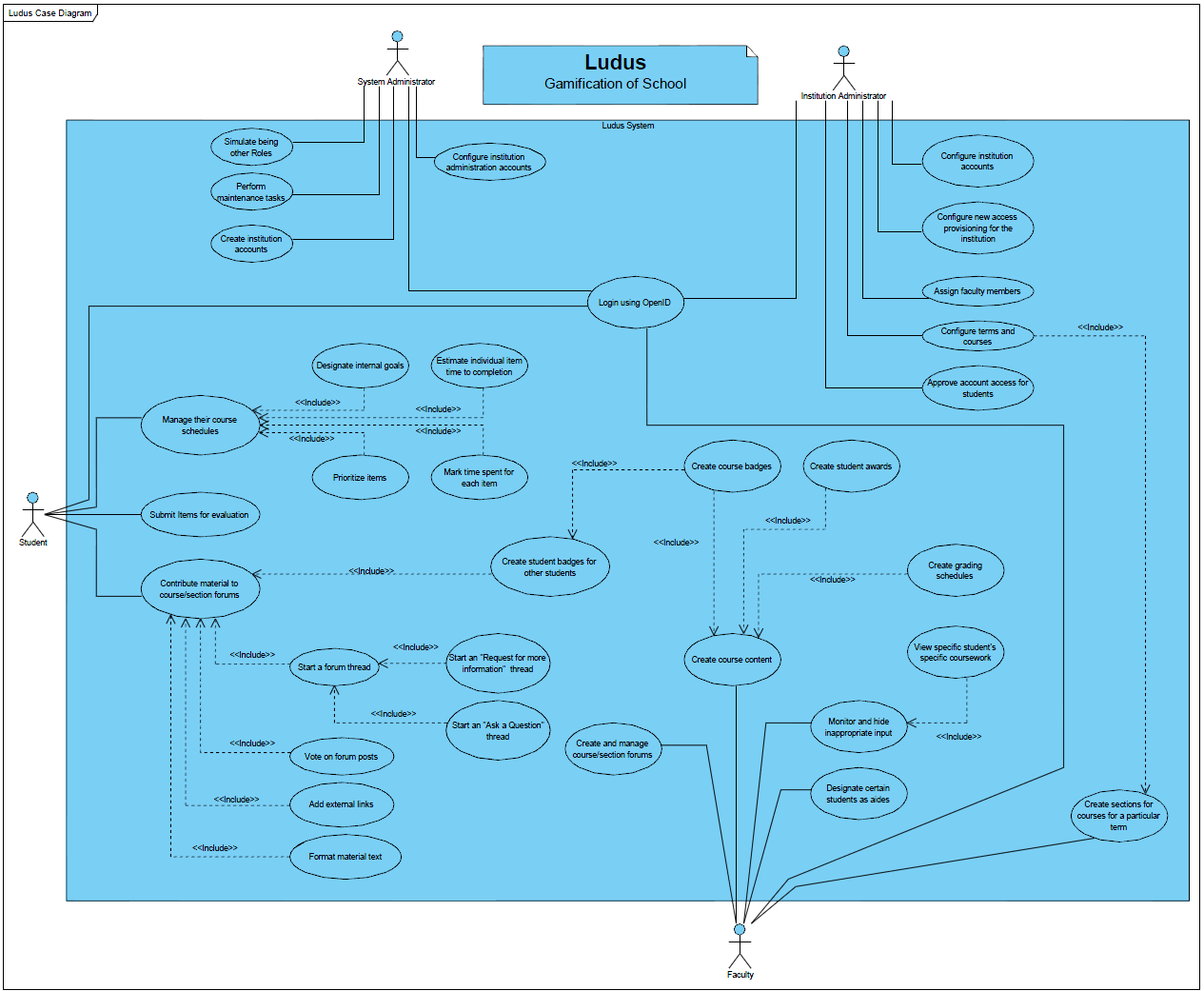
# 6. Other Requirements

## Appendix A: Glossary

Through this document, the term “activity” and “activities” shall be used to indicate classroom-based activities, such as a lecture, homework, quiz, examination, or project. Individual use of these terms indicates that the narrative applies only to those types noted, and not to the other types.

This document uses the name Ludus, and the term “system” interchangeably.

## Appendix B: Use-Case Diagram



## D:\Downloads\class.pngAppendix C: Class Diagram