

Summer 2017

EDUFORM

Olaf Broms
Dakota State University

Follow this and additional works at: <https://scholar.dsu.edu/theses>



Part of the [Databases and Information Systems Commons](#)

Recommended Citation

Broms, Olaf, "EDUFORM" (2017). *Masters Theses & Doctoral Dissertations*. 323.
<https://scholar.dsu.edu/theses/323>

This Thesis is brought to you for free and open access by Beadle Scholar. It has been accepted for inclusion in Masters Theses & Doctoral Dissertations by an authorized administrator of Beadle Scholar. For more information, please contact repository@dsu.edu.



EDUFORUM

A graduate project submitted to Dakota State University in partial fulfillment of
the requirements for the degree of

Master of Science

in

Information Systems

Summer 2017

By

Olaf Broms

Project Committee:

Dr. David Bishop
Dr. Christopher Olson
Mr. Robert Pope



DAKOTA STATE

MSIS Project
Project Idea Approval Form
(Form #1)

Important: Your project idea must be reviewed and approved by your project supervisor. (See Project Guidelines on grad office website - current students link for detailed description of the project)

Student Name: Olaf Broms Expected Graduation Date: Summer 2017

Faculty Project Supervisor: Dave Bishop D.Sc.

Master's Project Working Title: EduForum

Description of Your Idea: Maximum one page description/discussion of your idea, including proposed deliverables and a preliminary timeline for completion of the plan.

DO NOT WRITE YOUR IDEA HERE. Attach a word-processed page describing the idea as described above.

Students must bring the original form to the Graduate Programs Office. Graduate Office personnel will make copies and return them to your project supervisor and the MSIS Program Coordinator. You should retain a copy for your files.

Approvals/Signatures:

Student: 

Date: 30 Nov 16

Faculty supervisor: Dave Bishop

Date: 12/01/2016

MSIS Coordinator: Ronghua Shan

Date: 12/02/2016

Original to Office of Graduate Studies and Research



MSIS
PROJECT APPROVAL FORM
(Form #3)

Student Name: Broms, Olaf A.

Master's Project Title: EduForum

Signature Approval: 

Faculty supervisor: Dave Bishop

Date: August 17, 2017

Committee member: Christopher Olson

Date: August 17, 2017

Committee member: Robert Pope

Date: 8/21/2017

DocuSigned by:
Robert Pope
E5C5E77CC14E416...

ACKNOWLEDGMENT

I would like to acknowledge those people and organizations whom had a part in the formulation and oversight in the development of EduForum.

I would like to thank Dr. David Bishop for his patience and guidance in directing the project throughout its course

I would like to thank Dr. Christopher Olson for serving as a member of the project committee.

I would like to thank Mr. Robert Pope for serving as a member of the project committee and for providing guidance throughout the project.

I would also like to thank the Online Learning Consortium for allowing me to realize a need in the market that lead to the creation of the EduForum application.

Abstract

EduForum is a project that has been constructed to fulfil a niche in the learning management system market. The current learning management system market is comprised of mainly large, well-funded, and highly entrenched offerings. The trend in the market has been shifting recently away from some of these legacy offerings towards some of their newer competitors. This trend holds true in the for-profit market, but no true competitor has emerged to challenge the market's largest open-source offering, Moodle.

EduForum hopes to be the offering to fill this role by using a number of new methodologies and technologies to create a robust and user friendly offering. EduForum has been constructed to be easy to develop new features for, to be user-friendly, and to meet the learning needs of all potential users.


The project was constructed during the course of two semesters at Dakota State University. The project was broken into distinct phases that encompassed the conception, project planning, and project implementation. The tooling of the project was designed to allow for the easy development of new features so as to facilitate its easy expansion. The development of the project was further broken down into development sprints so as to take advantage of an agile methodology. Utilizing this methodology during development allowed for project development focus to methodically shift focus to respond to perceived issues or to focus in a different direction. Additionally, "Developing in short, time-boxed iterations and demonstrating to stakeholders early and often enables teams and stakeholders to agree when the product has met each of the needs" (Ekas, 2012)

4 DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution. |

Signed,

A handwritten signature in black ink, appearing to read "S. Brown", written over a horizontal line.

<Student name>

TABLE OF CONTENTS

ACKNOWLEDGMENT	III
TABLE OF CONTENTS	VI
LIST OF FIGURES	VIII
INTRODUCTION	1
BACKGROUND OF THE PROBLEM.....	1
STATEMENT OF THE PROBLEM.....	1
OBJECTIVES OF THE PROJECT	1
LITERATURE REVIEW	3
SYSTEM DESIGN.....	5
USE CASE 1 CREATE INSTITUTION	10
CHARACTERISTIC INFORMATION	10
<i>Brief description</i>	10
<i>Primary Actor</i>	10
<i>Secondary Actor</i>	10
<i>Stakeholders</i>	10
<i>Trigger</i>	10
<i>Preconditions</i>	11
<i>Guarantees</i>	11
MAIN SUCCESS SCENARIO	11
EXTENSIONS	11
USE CASE 2 CREATE COURSE.....	11
CHARACTERISTIC INFORMATION	11
<i>Brief description</i>	11
<i>Primary Actor</i>	11
<i>Secondary Actor</i>	11
<i>Stakeholders</i>	11
<i>Trigger</i>	11
<i>Preconditions</i>	11
<i>Guarantees</i>	11
MAIN SUCCESS SCENARIO	12
EXTENSIONS	12
USE CASE 3 CREATE ADMINISTRATOR.....	12
CHARACTERISTIC INFORMATION	12
<i>Brief description</i>	12
<i>Primary Actor</i>	12
<i>Secondary Actor</i>	12
<i>Stakeholders</i>	12
<i>Trigger</i>	12
<i>Preconditions</i>	12
<i>Guarantees</i>	12
MAIN SUCCESS SCENARIO	12
EXTENSIONS	13
USE CASE 4 CREATE INSTRUCTOR.....	13
CHARACTERISTIC INFORMATION	13
<i>Brief description</i>	13
<i>Primary Actor</i>	13
<i>Secondary Actor</i>	13
<i>Stakeholders</i>	13
<i>Trigger</i>	13
<i>Preconditions</i>	13
<i>Guarantees</i>	13
MAIN SUCCESS SCENARIO	13
EXTENSIONS	13

USE CASE 5 CREATE STUDENT	14
CHARACTERISTIC INFORMATION	14
<i>Brief description</i>	14
<i>Primary Actor</i>	14
<i>Secondary Actor</i>	14
<i>Stakeholders</i>	14
<i>Trigger</i>	14
<i>Preconditions</i>	14
<i>Guarantees</i>	14
MAIN SUCCESS SCENARIO	14
EXTENSIONS	14
USE CASE 6 CREATE ASSIGNMENT	15
CHARACTERISTIC INFORMATION	15
<i>Brief description</i>	15
<i>Primary Actor</i>	15
<i>Secondary Actor</i>	15
<i>Stakeholders</i>	15
<i>Trigger</i>	15
<i>Preconditions</i>	15
<i>Guarantees</i>	15
MAIN SUCCESS SCENARIO	15
EXTENSIONS	15
<i>USE CASE 7 Maintain Administrator Information</i>	16
<i>Secondary Actor</i>	16
<i>USE CASE 8 Maintain Instructor Information</i>	18
<i>Secondary Actor</i>	18
<i>USE CASE 9 Maintain Student Information</i>	20
<i>Secondary Actor</i>	20
<i>USE CASE 10 Maintain Institution Information</i>	22
<i>Secondary Actor</i>	22
<i>USE CASE 11 Maintain Course Information</i>	24
<i>Secondary Actor</i>	25
<i>USE CASE 12 Maintain Assignment Information</i>	27
<i>Secondary Actor</i>	27
CASE STUDY	30
CONCLUSIONS	31
REFERENCES	32
APPENDIX A: USERS' MANUAL	33
GENERAL SYSTEM INFORMATION	34
PLATFORM ADMINISTRATION	38
INSITUTIONAL ADMINISTRATION	41
COURSE ADMINISTRATION	44
STUDENT USE	46
APPENDIX B: SYSTEM TECHNICAL DOCUMENTATION	49

LIST OF FIGURES

Figure 1. USE-CASE DIAGRAM	Error! Bookmark not defined.
Figure 2. PROJECT GANTT CHART.....	41
Figure 3. WORK-BREAKDOWN STRUCTURE	42
Figure 4. ENTITY-RELATIONSHIP DIAGRAM	43
Figure 5. DATABASE TABLES	44

CHAPTER 1

INTRODUCTION

Background of the Problem

The Learning Management System(LMS) sector is a market that is growing rapidly as the idea of online education becomes a more mainstream approach to education. The majority of LMS platforms are mostly comprised of for-profit offerings. There are open source solutions, but these are mostly comprised of older and more monolithic platforms. It was also found that of an institutions budget, “Learning Management Systems account for 38% of the average learning technology budget” (Simpson, 2017).

Statement of the problem

The older and more expensive offerings in the LMS market offer a challenge for new users in both their use, and the capability to afford licensing fees. These painful options are often exacerbated by the nature of monolithic systems; they are built upon massive legacy codebases that can cause endless issues for the end user. Inside Higher Ed recently conducted a study that found that many institutions are dropping adoption of the older LMS offerings and seeking newer technology from fresher competitors. “The two biggest drivers of those competitions, he said, are the existence of Canvas -- the only one of the leading LMSs "that wasn't designed in the late 1990s/early 2000s" -- and expectations from consumers who are increasingly frustrated by some of the poor user experiences in many LMSs” (Lederman, 2017). There are very few learning platforms that offer an easy to use, reliable, and low cost option. EduForum plans to fill this gap in the market.

Objectives of the project

EduForum is hoped to fill a niche in the LMS market by offering a solution to a problem that is not widely addressed, which is to enable online learning for those who are least able to afford it. This is not to exclude corporate or academic institutional training, but

by using open source technologies and by keeping the project itself open source, a light weight and easily to use LMS can be accessible to the many. This project will also further the development teams experience with configuration of open source platforms and development using state of the art technologies. A user manual is attached in Appendix A which will contain version 1 EduForum user interfaces. Appendix B will show code samples and the version 1 data model as well as screenshots of code samples and some of the platforms used in creation of EduForum.

CHAPTER 2

LITERATURE REVIEW

Current Market

Learning management systems, as a software offering, have been around since the 1970s. The current market is comprised of a great variety of offerings, based on a number of different platforms, and having a wide array of associated functionality. The learning management system as a platform has begun to diverge into an array of offerings to include the learning management system, content creation systems, data stores for developed content, and others as the platforms mature.

The current LMS market is dominated by a number of paid-for offerings, rounded out by the open-source Moodle LMS. The paid-for learning management system market is dominated by a few high-profile LM systems such as BlackBoard and D2L with Moodle being the most popular offering in the open-source sector.

As an open source offering, Moodle is the learning management system that will most directly compare to EduForum. Moodle is truly a giant in the open source learning management system market. Moodle was originally founded in 2002 and development continues to this day under the direction of Moodle HQ. As a product of its age and adoption there are a number of plug-ins that may be added to the core product to enhance its capability. Moodle's age and customizability while being a sign of its popularity, have lead to its widespread adoption. Due to Moodle's adoption and the fact that many academic institutions may not have the financial capability to update platforms regularly has led Moodle to maintain a large legacy code base. This legacy codebase, while allowing underfunded institutions to maintain an existing learning management system has lead to a large amount of technical debt within the platform. Additionally, Moodle's lack of JavaScript support and older PHP version support ensures that its performance will be substandard as compared to modern applications. As part of the research it was found that "Moodle is community-supported, which leads Moodle to lack a clear systematic structure and a set of well-kept documentation" (Wu, 2017). Another drawback is its own customizability. While normally

viewed as a great benefit, this customizability leads non-technically inclined users to either overlook Moodle, or to hire an outside service to set up their instance of Moodle.

EduForum fits in this market as being a Learning Management System that while being open source, will be designed to circumvent the usability problems associated with the use of Moodle. To accomplish this the user experience will be streamlined and content management and navigation will be made priorities throughout the build process, by using methodologies and technology not available to Moodle. Market research has found that “61 % of companies plan on replacing their learning platforms within the next 18 months” (Bersin, 2014). This demonstrated lack of satisfaction with existing systems should prove a boon to EduForum’s market prospects.

CHAPTER 3

SYSTEM DESIGN

Project Phases

The EduForum project was planned and executed throughout a number of phases which incrementally moved the system towards being a completed product. These phases were then grouped into the two active semesters spent planning and executing the project plan. The project idea phase laid the groundwork for the project planning phase. The information and plans brought together during this phase allowed for the planning of the development phase. The development phase then consisted of the actual platform development.

The project idea phase was the starting phase of the project and led to the development of the formal project idea and initial plan outline. To conceptualize EduForum, current needs related to Online Learning Consortium business problems were analyzed. After evaluating a number of options, it was found that the in-place learning management system caused a number of issues for employees at the Online Learning Consortium. Research was then conducted into existing offerings, the learning management system market, and into system complexity. This research pointed towards there being space in the market for a new, user-friendly, open source learning management system. Once this fact was realized, project needs were broken down into weekly goals in the form of an initial project planning phase roadmap.

Following the project idea phase, the project moved into the actual planning phase. The planning phase was broken down into 15 weekly milestones that when completed, would allow for the formation of a development plan. These milestones were comprised of further market and competitor research, tentative deliverables declarations, use cases, risk and quality management plans, work scheduling, data modeling, interface design, and wireframing. During the project planning phase it was realized that a project of the proposed magnitude may not be feasible, and so plans were made to secure additional developer assistance if the project failed to meet development milestones.

The project development planning phase was made possible by the information gathered during the project planning phase. The project development phase was broken into six 2-week development sprints, to take advantage of an agile project management methodology. This methodology would allow for sprint-based project iterations, designed to deliver the basic product and improve it throughout the development phase. The initial sprint differed from the following sprints as it comprised the bulk of tooling set up. These tools consisted of a deployment pipeline, project management software, server configuration, automated test configuration, and web host acquisition. Only new features were considered for the first three sprints with the remaining three sprints being left open so as to flexibly adjust to problems and to iteratively update existing features.

Having the development plan in place, the second semester of the project began with the start of the development phase. The development phase of the project was planned to use the PHP framework Laravel for server side assets, the JavaScript framework Vue.JS for client side assets, git version control, automated testing to ensure code quality, and continuous integration to save development time on deployments. Due to lack of development team experience using some of the technologies mentioned, there was a slight amount of delay during the initial sprints. Once the team gained the proper experience, artifact deliverability increased greatly. It was also realized during the development phase that changes to the data model would have to be made to account for oversights during the project planning phase. These were minor setbacks though, that allowed the team to realize a further benefit of the agile methodology, in that it would allow the team to more rapidly pivot in a new direction to address these oversights.

Sprint Objectives

Sprint 1 – 15 May – 29 May

- Project Setup and Configuration.
- Public facing Repository Creation.
- Development and Data Storage preparation.
- Employee Project Familiarization.
- Development Sprint Planning.
- Ensure Completeness of Project Setup.
- Functionality to allow for Users Creation and User Management.

Project Tooling and Initial Development – Project Management, Hosting, Domains, Database Creation, Git Setup, Build Tools, Test Framework Implementation, Deployment Pipeline Setup

Sprint 2 – 29 May – 15 Jun

- Functionality to allow for Institution Management.
- Functionality to allow for Course Management.

Sprint 2 Comments – Finish any additional tooling, demo Functionality only build, test initial build, deploy initial build, Development plan re-visitation based off first 2 sprints observations

Sprint 3 – 15 Jun – 30 Jun

- Functionality to allow for Schedule Management.
- Address any bypassed, lower level functionality falling within the previously addressed functional areas.
- Finish any incomplete functionality from the previous sprint.
- Readdress project milestones and follow-on Sprint Planning

Sprint 3 Comments – Build off previous build sprint. Re-evaluate Development Plan. Adjust workflow where shortcomings arise. Acquire additional resources based off any shortfalls in previous sprints

Sprint 4 – 1 Jul – 15 Jul

- Readdress previously added functionality.
- Streamline user registration flow.

Sprint 4 Comments – Build off previous build sprint. Re-evaluate Development Plan to account for issues or new direction in priority. User flow when registering for platform, institution, and courses.

Sprint 5 – 15 Jul – 30 Jul

- Readdress previously added functionality.
- Ensure completeness of user roles and access control.
- Streamline administration flow for administering institutions.
- Complete course level administration and use

Sprint 5 Comments – Build off previous build sprint. Clean up administration and role systems. Finish development of course level usage

Sprint 6 – 1 Aug – 11 Aug

- Readdress previously added functionality.
- Project Clean-up
- Project follow up reporting

Sprint 5 Comments – Build off previous build sprint. Address user interface, styling and any previously recognized bugs

Platform Selection

There are a number of factors that went into the selection of the platforms used in the project. Of great importance was the selection of Laravel as the primary product framework. This was selected due to the development teams experience with the platform, as well as the fact that Laravel is comprised of a number of features that allow fast development of web applications. For database interaction, Laravel includes the Eloquent object relational mapper. Eloquent simplifies database interactions, so the team would spend less time constructing stored procedures or PDO requests. Laravel also makes it trivial to create models, controllers, database seeders, and migrations.

Vue.js was selected as the front-end framework because of its reputation as being easy to incrementally adopt. This was a very positive aspect of the framework, as the team having no experience with it, could expand its use once experience had been gained. The team at Gitlab recently adopted Vue.js and were happy to find that “ Its source code is very readable, and the documentation is the only tutorial you'll ever need”(Schatz, 2016). Additionally, Vue.js was appealing because of the ease in which usable components can be constructed with varied external data.

Version control was imperative for this project, as with most others, as it allowed the team to rapidly deploy changes as well as keep track of past changes. The ability to roll back code issues proved useful throughout the project.

The automated testing and continuous integration pipeline proved to be a great time saver over the course of the project as well. Once initially configured any changes sent to the github repository were automatically tested. Upon completion, they were sent to the live server and were made live.

Project Use Cases

The modeling for EduForum was conceptualized based upon the analyzation of existing platforms and team experience in the utilization of similar systems. The system was then broken down into a number of use cases, signifying actions various users may take when utilizing the system. These are detailed later in this section. Further these user cases were then summarized in a use case diagram, so as to better conceptualize the various user interactions with the system. This is detailed in figure 1.

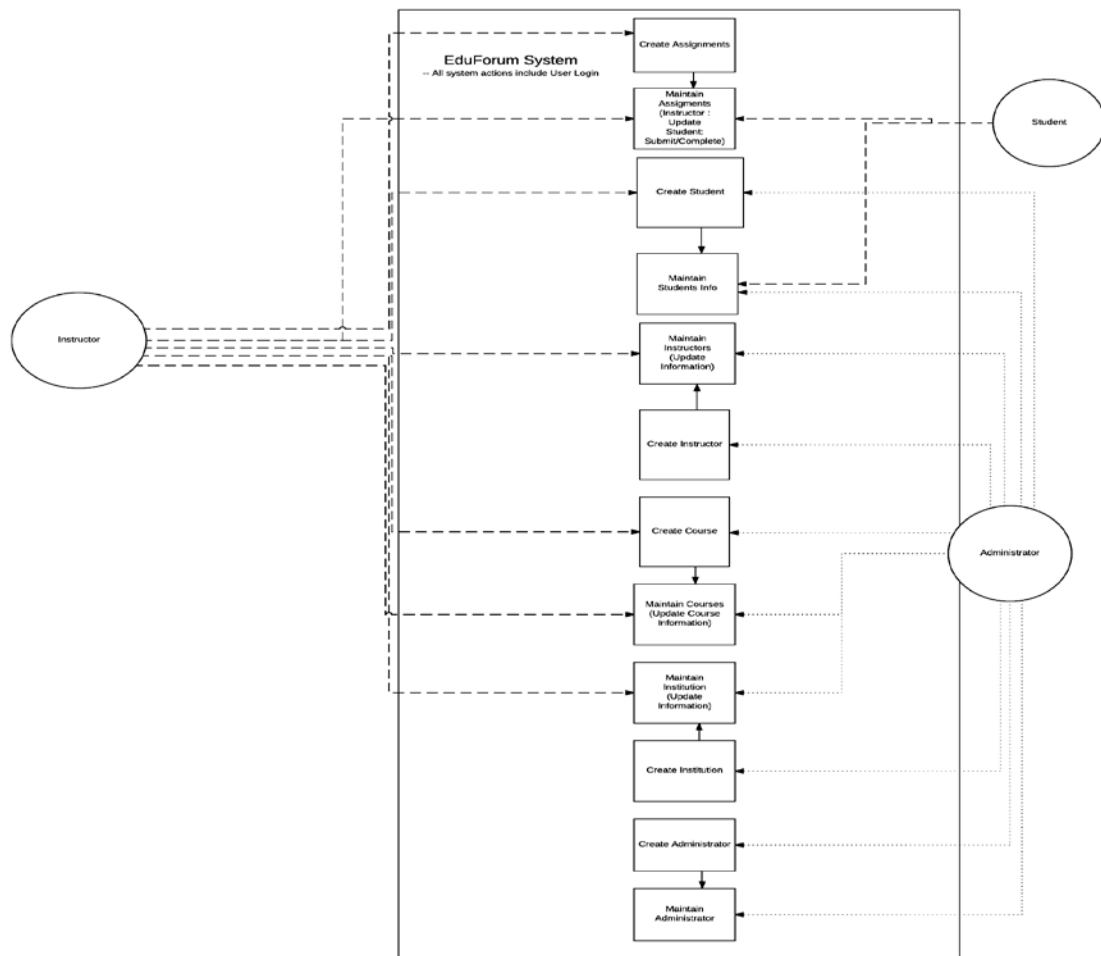


Figure 1- Use Case Diagram

The individual use cases taken into consideration are detailed in the following section.

USE CASE 1 CREATE INSTITUTION

CHARACTERISTIC INFORMATION

Brief description

This use case covers the act of creating an Institution

Primary Actor

Administrator

Secondary Actor

EduForum System

Stakeholders

Administrator

Institution

Trigger

A new institution needs to offer online courses

Preconditions

Admin is logged into the EduForum system and their credentials have been validated

Guarantees**Success End Condition**

Institution is created

Failed End Condition

Institution not created

MAIN SUCCESS SCENARIO

- 1a. The Admin accesses the EduForum system institution creation section
- 2a. The Admin enter Institution Information
- 3a. The system validates Institution information
- 4a. The Institution is created

EXTENSIONS

- *a. The EduForum system suffers a loss of service
 - a1. The Institutions is not created [fail]

- 2a. The Institution information is incorrect
 - 2a1. The Admin corrects the error

- 3a. Institution data is invalid
 - 4a1. The system notifies the Admin that data is invalid [repeat at 2]

USE CASE 2 CREATE COURSE**CHARACTERISTIC INFORMATION****Brief description**

This use case covers the act of creating a course

Primary Actor

Administrator / Instructor

Secondary Actor

EduForum System

Stakeholders

Administrator

Instructor

Institution

Trigger

A new course is to be offered

Preconditions

Admin is logged into the EduForum system and their credentials have been validated

Guarantees**Success End Condition**

Course is created

Failed End Condition

Course not created

MAIN SUCCESS SCENARIO

- 1a. The Admin / Instructor accesses the EduForum system course creation section
- 2a. The Admin / Instructor enters course Information
- 3a. The system validates course information
- 4a. The course is created

EXTENSIONS

- *a. The EduForum system suffers a loss of service
 - a1. The course is not created [fail]

- 2a. The course information is incorrect
 - 2a1. The Admin corrects the error

- 3a. Course data is invalid
 - 4a1. The system notifies the Admin that data is invalid [repeat at 2]

USE CASE 3 CREATE ADMINISTRATOR**CHARACTERISTIC INFORMATION****Brief description**

This use case covers the act of creating an administrator

Primary Actor

Administrator

Secondary Actor

EduForum System

Stakeholders

Administrator

Institution

Trigger

A new Administrator / account is needed

Preconditions

Admin has acquired the necessary software or ability to access the software

Guarantees**Success End Condition**

Administrator is created

Failed End Condition

Administrator not created

MAIN SUCCESS SCENARIO

- 1a. The Admin sets up and installs software in appropriate location
- 2a. The Admin registers as the system admin user
- 3a. The system validates Admin information
- 4a. The Admin is created

EXTENSIONS

- *a. The EduForum system suffers a loss of service
 - a1. The Admin is not created [fail]

- 1a. The installation / setup is unsuccessful
 - 1a1. The Admin corrects the error[repeat]

- 2a. The Admin information is incorrect
 - 2a1. The Admin corrects the error

- 3a. Admin data is invalid
 - 4a1. The system notifies the Admin that data is invalid [repeat at 2]

USE CASE 4 CREATE INSTRUCTOR**CHARACTERISTIC INFORMATION****Brief description**

This use case covers the act of creating an instructor

Primary Actor

Administrator

Secondary Actor

EduForum System

Stakeholders

Administrator

Institution

Trigger

A new Instructor / account is needed

Preconditions

Admin is logged into the EduForum system and their credentials have been validated

Guarantees**Success End Condition**

Instructor is created

Failed End Condition

Instructor not created

MAIN SUCCESS SCENARIO

- 1a. The Admin accesses the EduForum system user creation section
- 2a. The Admin enters the user's permissions and information
- 3a. The system validates user information
- 4a. The Instructor is created

EXTENSIONS

- *a. The EduForum system suffers a loss of service
 - a1. The user is not created [fail]

- 2a. The user information is incorrect
 - 2a1. The Admin corrects the error

- 3a. Admin data is invalid
 - 4a1. The system notifies the Admin that data is invalid [repeat at 2]

USE CASE 5 CREATE STUDENT

CHARACTERISTIC INFORMATION

Brief description

This use case covers the act of creating a student

Primary Actor

Administrator / Instructor

Secondary Actor

EduForum System

Stakeholders

Administrator

Instructor

Institution

Trigger

A new student / account is needed

Preconditions

Admin / Instructor is logged into the EduForum system and their credentials have been validated

Guarantees

Success End Condition

Student is created

Failed End Condition

Student not created

MAIN SUCCESS SCENARIO

- 1a. The Administrator / Instructor accesses the EduForum system user creation section
- 2a. The Administrator / Instructor enters the user's permissions and information
- 3a. The system validates user information
- 4a. The student is created

EXTENSIONS

- *a. The EduForum system suffers a loss of service
 - a1. The user is not created [fail]

- 2a. The user information is incorrect

2a1. The Administrator / Instructor corrects the error

3a. Student data is invalid

4a1. The system notifies the Administrator / Instructor that data is invalid [repeat at 2]

USE CASE 6 CREATE ASSIGNMENT

CHARACTERISTIC INFORMATION

Brief description

This use case covers the act of creating an assignment

Primary Actor

Administrator / Instructor

Secondary Actor

EduForum System

Stakeholders

Administrator

Instructor

Institution

Trigger

A new student / account is needed

Preconditions

Instructor is logged into the EduForum system and their credentials have been validated

Guarantees

Success End Condition

Assignment is created

Failed End Condition

Assignment not created

MAIN SUCCESS SCENARIO

1a. The Instructor accesses the EduForum system assignment section and selects to create a new assignment

2a. The Instructor enters the assignment information and assignment content.

3a. The system validates the assignment information and checks data sources

4a. The assignment is created

EXTENSIONS

*a. The EduForum system suffers a loss of service

a1. The assignment is not created [fail]

2a. The assignment information is incorrect

2a1. The Instructor corrects the error

3a. Assignment data is invalid

3a1. The system notifies the Instructor that data is invalid [repeat at 2]

3b. External data source is invalid

3b1. The system notifies the Instructor that external data is invalid [repeat at 2]

USE CASE 7 Maintain Administrator Information

CHARACTERISTIC INFORMATION

Brief description

This use case allows the Administrator to update administrator information.

Primary Actor

Administrator

Secondary Actor

EduForum System

Stakeholders

Administrator
Institution

Trigger

Administrator accesses EduForum user section and selects to update an existing user

Preconditions

Administrator is logged in and has permissions to update the selected user

Guarantees

Success End Condition

Administrator user is updated

Failed End Condition

Nothing happens

MAIN SUCCESS SCENARIO

1a. The system presents the Administrator with the user update options

2ab. The Administrator selects Administrator to update and updates Administrator information

3a. The Administrator submits updated user information.

4a. The system validates Administrator information per Administrator information rules.

5. The system saves Administrator information and presents the Administrator with a confirmation.

EXTENSIONS

*a. The EduForum system suffers a loss of service

a1. The Administrator is not updated[fail]

*b. Administrator decides to quit:

*b1. System asks the Administrator to save changes (if any)

*b1b. Administrator chooses to save:

System saves changes and exits [success]

*b1b. Administrator chooses to quit:

System discards any intermediate data and exits [fail]

1a. The Administrator selects to 'Update Administrator information':

1a1. The Administrator selects an existing Administrator.

1a2. System ensures selected Administrator is a valid user.

1a2a. Administrator not a valid user

System notifies Administrator [repeat]

1a2b. Administrator is valid:

System presents Administrator information.

Administrator provides updated information.

System validates Administrator info per Administrator info rules.

Invalid Administrator information:

System notifies Administrator [repeat]

System saves Administrator info, presents Administrator with

confirmation.

2d. The Administrator requests to 'Delete Administrator':

2d1. The Administrator selects an existing Administrator.

2d2. System ensures Administrator is a valid user.

2d2a. Administrator not valid

System notifies Administrator [repeat]

2d2b. Administrator is valid:

System presents Administrator information.

Administrator confirms Administrator is correct and selects to delete

the Administrator.

The system confirms that Administrator wants to delete Administrator.

Cancel deletes operation:

System aborts delete and notifies Administrator [repeat]

System deletes Administrator and presents Administrator with confirmation.

- 3a. Invalid Administrator information:
 - 3a1. System notifies Administrator [repeat]

USE CASE 8 Maintain Instructor Information

CHARACTERISTIC INFORMATION

Brief description

This use case allows the Administrator / Instructor to update instructor information.

Primary Actor

Administrator
Instructor

Secondary Actor

EduForum System

Stakeholders

Administrator
Instructor
Institution

Trigger

Administrator / Instructor accesses EduForum user section and selects to update an existing user

Preconditions

Administrator / Instructor is logged in and has permissions to update or delete the selected user

Guarantees

Success End Condition

Instructor user is updated

Failed End Condition

Nothing happens

MAIN SUCCESS SCENARIO

- 1a. The system presents the Administrator / Instructor with the user update options

- 2ab. The Administrator / Instructor selects Instructor to update and updates Instructor information
- 3a. The Administrator / Instructor submits updated user information.
- 4a. The system validates Instructor information per Instructor information rules.
- 5. The system saves Instructor information and presents the Administrator / Instructor with a confirmation.

EXTENSIONS

- *a. The EduForum system suffers a loss of service
 - a1. The Instructor is not updated[fail]

- *b. Administrator / Instructor decides to quit:
 - *b1. System asks the Administrator / Instructor to save changes (if any)
 - *b1b. Administrator / Instructor chooses to save:
 - System saves changes and exits [success]
 - *b1b. Administrator / Instructor chooses to quit:
 - System discards any intermediate data and exits [fail]

- 1a. The Administrator / Instructor selects to 'Update Instructor information':
 - 1a1. The Administrator / Instructor selects an existing Instructor.
 - 1a2. System ensures selected Instructor is a valid user.
 - 1a2a. Instructor not a valid user
 - System notifies Administrator / Instructor [repeat]
 - 1a2b. Instructor is valid:
 - System presents Instructor information.
 - Administrator / Instructor provides updated information.
 - System validates Instructor info per Instructor info rules.
 - Invalid Instructor information:
 - System notifies Administrator / Instructor [repeat]
 - System saves Instructor info, presents Administrator / Instructor with confirmation.

- 2d. The Administrator selects to 'Delete Administrator':
 - 2d1. The Administrator selects an existing Instructor.
 - 2d2. System ensures Instructor is a valid user.
 - 2d2a. Instructor not valid
 - System notifies Administrator / Instructor [repeat]
 - 2d2b. Instructor is valid:
 - System presents Instructor information.
 - Administrator confirms Instructor is correct and selects to delete the Instructor.

The system confirms that Administrator wants to delete Instructor.

Cancel deletes operation:

System aborts delete and notifies Administrator [repeat]

System deletes Instructor and presents Administrator with

confirmation.

3a. Invalid Instructor information:

3a1. System notifies Administrator / Instructor [repeat]

Instructor Information Rules

1	For additions, assign a unique ID.
2	Instructor may update their information only.
3	Administrator may update Instructor information or delete instructor

USE CASE 9 Maintain Student Information

CHARACTERISTIC INFORMATION

Brief description

This use case allows the Administrator / Instructor / Student to update student information.

Primary Actor

Administrator

Instructor

Student

Secondary Actor

EduForum System

Stakeholders

Administrator

Instructor

Student

Institution

Trigger

Administrator / Instructor accesses EduForum user section and selects to update an existing user

Preconditions

Administrator / Instructor is logged in and has permissions to update or delete the selected user

Guarantees*Success End Condition*

Instructor user is updated

Failed End Condition

Nothing happens

MAIN SUCCESS SCENARIO

- 1a. The system presents the Administrator / Instructor / Student with the user update options
- 2ab. The Administrator / Instructor / Student selects student to update and updates user information
- 3a. The Administrator / Instructor / Student submits updated user information.
- 4a. The system validates user information per Instructor information rules.
5. The system saves Instructor information and presents the Administrator / Instructor with a confirmation.

EXTENSIONS

- *a. The EduForum system suffers a loss of service
 - a1. The Instructor is not updated[fail]

- *b. Administrator / Instructor / Student decides to quit:
 - *b1. System asks the Administrator / Instructor / Student to save changes (if any)
 - *b1b. Administrator / Instructor / Student chooses to save:
 - System saves changes and exits [success]
 - *b1b. Administrator / Instructor / Student chooses to quit:
 - System discards any intermediate data and exits [fail]

- 1a. The Administrator / Instructor / Student selects to 'Update Student information':
 - 1a1. The Administrator / Instructor / Student selects an existing Student.
 - 1a2. System ensures selected Student is a valid user.
 - 1a2a. Student not a valid user
 - System notifies Administrator / Instructor / Student [repeat]
 - 1a2b. Student is valid:
 - System presents Student information.
 - Administrator / Instructor / Student provides updated information.
 - System validates Instructor info per Instructor info rules.
 - Invalid Instructor information:
 - System notifies Administrator / Instructor [repeat]
 - System saves Instructor info, presents Administrator / Instructor with confirmation.

- 2d. The Administrator selects to 'Delete Student:
- 2d1. The Administrator selects an existing Student.
 - 2d2. System ensures Student is a valid user.
 - 2d2a. Student not valid
 - System notifies Administrator [repeat]
 - 2d2b. Student is valid:
 - System presents Student information.
 - Administrator confirms Student is correct and selects to delete the Student.
 - The system confirms that Administrator wants to delete Student.
 - Cancel deletes operation:
 - System aborts delete and notifies Administrator [repeat]
 - System deletes Student and presents Administrator with confirmation.

- 3a. Invalid Student information:
- 3a1. System notifies Administrator / Instructor / Student [repeat]

Student Information Rules

1	For additions, assign a unique ID.
2	Instructor may update student information or remove from course.
3	Administrator may update student information, remove from course, or delete student
4	Student may update Student personal information

USE CASE 10 Maintain Institution Information

CHARACTERISTIC INFORMATION

Brief description

This use case allows the Administrator to update Institution information.

Primary Actor

Administrator

Secondary Actor

EduForum System

Stakeholders

Administrator

Institution

Trigger

Administrator accesses EduForum user section and selects to update an existing Institution

Preconditions

Administrator is logged in and has permissions to update or delete the selected Institution

Guarantees

Success End Condition

Institution is updated

Failed End Condition

Nothing happens

MAIN SUCCESS SCENARIO

- 1a. The system presents the Administrator with the institution update options
- 2ab. The Administrator selects institution to update and updates institution information
- 3a. The Administrator submits updated institution information.
- 4a. The system validates institution information per institution information rules.
5. The system saves institution information and presents the Administrator with a confirmation.

EXTENSIONS

- *a. The EduForum system suffers a loss of service
 - a1. The institution is not updated[fail]

- *b. Administrator decides to quit:
 - *b1. System asks the Administrator to save changes (if any)
 - *b1b. Administrator chooses to save:
 - System saves changes and exits [success]
 - *b1b. Administrator chooses to quit:
 - System discards any intermediate data and exits [fail]

- 1a. The Administrator selects to 'Update Institution information':
 - 1a1. The Administrator selects an existing Institution.
 - 1a2. System ensures selected Institution is valid.
 - 1a2a. Institution is not valid
 - System notifies Administrator [repeat]
 - 1a2b. Institution is valid:
 - System presents Institution information.
 - Administrator provides updated information.
 - System validates Institution info per Institution info rules.

- Invalid Institution information:
 - System notifies Administrator [repeat]
 System saves Institution info, presents Administrator with confirmation.

- 2d. The Administrator selects to 'Delete Institution':
 - 2d1. The Administrator selects an existing Institution.
 - 2d2. System ensures Institution is valid.
 - 2d2a. Institution not valid
 - System notifies Administrator[repeat]
 - 2d2b. Institution is valid:
 - System presents Institution information.
 - Administrator confirms Institution is correct and selects to delete the Institution.
 - The system confirms that Administrator wants to delete Institution.
 - Cancel deletes operation:
 - System aborts delete and notifies Administrator [repeat]
 - System deletes Institution and presents Administrator with confirmation.

- 3a. Invalid Institution information:
 - 3a1. System notifies Administrator / Instructor / Student [repeat]

Institution Information Rules

1	Only Administrators may delete Institutions

USE CASE 11 Maintain Course Information

CHARACTERISTIC INFORMATION

Brief description

This use case allows the Administrator / Instructor to update course information.

Primary Actor

- Administrator
- Instructor

Secondary Actor

EduForum System

Stakeholders

Administrator

Instructor

Student

Institution

Trigger

Administrator / Instructor accesses EduForum user section and selects to update an existing course

Preconditions

Administrator / Instructor is logged in and has permissions to update or delete the selected course

Guarantees

Success End Condition

Course is updated

Failed End Condition

Nothing happens

MAIN SUCCESS SCENARIO

- 1a. The system presents the Administrator / Instructor with the course update options
- 2ab. The Administrator / Instructor selects course to update and updates course information
- 3a. The Administrator / Instructor submits updated course information.
- 4a. The system validates course information per course information rules.
5. The system saves course information and presents the Administrator / Instructor with a confirmation.

EXTENSIONS

- *a. The EduForum system suffers a loss of service
 - a1. The Course is not updated[fail]
- *b. Administrator / Instructor decides to quit:
 - *b1. System asks the Administrator / Instructor to save changes (if any)
 - *b1b. Administrator / Instructor chooses to save:
 - System saves changes and exits [success]
 - *b1b. Administrator / Instructor chooses to quit:
 - System discards any intermediate data and exits [fail]

- 1a. The Administrator / Instructor selects to ‘Update Course information’:
- 1a1. The Administrator / Instructor selects an existing course.
 - 1a2. System ensures selected course is valid.
 - 1a2a. Course not a valid user
System notifies Administrator / Instructor [repeat]
 - 1a2b. Course is valid:
System presents Course information.
Administrator / Instructor provides updated information.
System validates Course info per Course info rules.
Invalid Course information:
System notifies Administrator / Instructor [repeat]
System saves Course info, presents Administrator / Instructor with confirmation.
- 2d. The Administrator selects to ‘Delete Course’:
- 2d1. The Administrator selects an existing Course.
 - 2d2. System ensures Course is valid.
 - 2d2a. Course not valid
System notifies Administrator[repeat]
 - 2d2b. Course is valid:
System presents Course information.
Administrator confirms Course is correct and selects to delete the Course.
The system confirms that Administrator wants to delete Course.
Cancel deletes operation:
System aborts delete and notifies Administrator [repeat]
System deletes Course and presents Administrator with confirmation.
- 3a. Invalid Course information:
- 3a1. System notifies Administrator / Instructor / Student [repeat]

Student Information Rules

2	Instructor may update course information.
3	Administrator may update Course information, or delete Course.

USE CASE 12 Maintain Assignment Information

CHARACTERISTIC INFORMATION

Brief description

This use case allows the Instructor / Student to update assignment information.

Primary Actor

Instructor

Student

Secondary Actor

EduForum System

Stakeholders

Instructor

Student

Institution

Trigger

Instructor / Student accesses EduForum course section and selects to update an existing assignment

Preconditions

Instructor / Student is logged in and has permissions to update or delete the selected assignment

Guarantees

Success End Condition

Assignment is updated

Failed End Condition

Nothing happens

MAIN SUCCESS SCENARIO

- 1a. The system presents the Instructor / Student with the assignment update options
- 2ab. The Instructor / Student selects assignment to update and updates assignment information
- 3a. The Instructor / Student submits updated assignment information.
- 4a. The system validates assignment information per assignment information rules.
5. The system saves assignment information and presents the Instructor / Student with a confirmation.

EXTENSIONS

- *a. The EduForum system suffers a loss of service
 - a1. The Assignment is not updated[fail]

- *b. Instructor / Student decides to quit:
 - *b1. System asks the Instructor / Student to save changes (if any)
 - *b1b. Instructor / Student chooses to save:
 - System saves changes and exits [success]
 - *b1b. Instructor / Student chooses to quit:
 - System discards any intermediate data and exits [fail]

- 1a. The Instructor / Student selects to 'Update Assignment information':
 - 1a1. The Instructor / Student selects an existing Assignment.
 - 1a2. System ensures selected Assignment is valid.
 - 1a2a. Assignment not valid
 - System notifies Instructor / Student [repeat]
 - 1a2b. Assignment is valid:
 - System presents Assignment information.
 - Instructor / Student provides updated information.
 - System validates Assignment info per Assignment info rules.
 - Invalid Assignment information:
 - System notifies Instructor / Student [repeat]
 - System saves Assignment information, presents Instructor / Student with confirmation.

- 2d. The Instructor selects to 'Delete Assignment':
 - 2d1. The Instructor selects an existing Assignment.
 - 2d2. System ensures Assignment is valid.
 - 2d2a. Assignment not valid
 - System notifies Instructor[repeat]
 - 2d2b. Assignment is valid:
 - System presents Assignment information.
 - Instructor confirms Assignment is correct and selects to delete the Assignment.
 - The system confirms that Instructor wants to delete Assignment.
 - Cancel deletes operation:
 - System aborts delete and notifies Instructor[repeat]
 - System deletes Assignment and presents Instructor with confirmation.

- 3a. Invalid Assignment information:
 - 3a1. System notifies Instructor / Student [repeat]

Assignment Information Rules

1	Instructor may update Assignment information or delete Assignment.
---	--

2	Student may upload Assignments or edit Assignments

CHAPTER 4

CASE STUDY

Challenge

The challenges during the development of EduForum were many-fold. The primary issue facing its conception and development were the projects complexity. Learning management systems as an offering are specifically geared towards the creation and management of instructional materials and associated functionality. Their nature makes them very much like standard content management systems, however, there are a number of additional features that are needed to make the product a learning management system. These features add to the complexity of any dedicated learning management system.

Solution

The challenges inherent to learning management systems are difficult to overcome, but with the proper amount of planning, these challenges can be overcome. The process was begun with the research conducted during the initial idea phase research. This led to the understanding of learning management system platforms, so as to create a more robust project plan. The planning for the project and the project charter called for careful development and an easily manageable end user experience. These requirements led to increased difficulty during development, but were required for a finished product.

Result

The result has fared well, but the project development also proved to be a challenge due to the nature of the project. Careful planning, data modeling and development allowed the project to proceed and be successful, however to achieve parity with successful market offerings, more time must be spent in development of additional features.

CHAPTER 5

CONCLUSIONS

At the outset of the project it was realized that there was an opening in the market for a lightweight, open-source learning management system. The initial project plan was put together to assess the needs for the project as well as laying the groundwork for the implementation phase of the project. The project plan made clear the resources and information needed to proceed, and with this information in hand the project development phase commenced. The objectives in the development of the EduForum system were two-fold. First to develop a useful product, that could be potentially be a marketable product. Secondly, this project, and the specific software tools used, were selected to give the development team a more in depth understanding of modern development practices and software. The completed system is useful and with more development time spent towards polish, could be market feasible. The largest gain while developing this project, has been the more thorough understanding of project management processes and the increased familiarity with continuous integration, automated testing, JavaScript frameworks, and modern web development.

For EduForum to be a marketable product, there are a number of new features that are needed to bring it to parity with some of its more highly funded competitors. Additionally, further code reviews, bug fixes, and some minor refactoring should take place to make using EduForum a more user-friendly experience. Consulting a well-qualified user experience developer would also be helpful in making the visual experience more pleasing to users.

REFERENCES

- Bersin, J. (2014, August 28). *The Red Hot Market for Learning Management Systems*. Retrieved from <https://www.forbes.com>:
<https://www.forbes.com/sites/joshbersin/2014/08/28/the-red-hot-market-for-learning-technology-platforms/#23468ac447bb>
- Ekas, L. (2012, 06 29). *Five benefits of agile practices in software-intensive systems development*. Retrieved from IBM:
https://www.ibm.com/developerworks/community/blogs/invisiblethread/entry/five_benefits_of_agile_practices?lang=en
- Kats, Y. (2010). Learning Management System Technologies for Online Teaching : Tools and Applications. In Y. Kats. Information Sciences Reference.
- Lederman, D. (2017, May 17). *The Learning Management Landscape*. Retrieved from Inside Higher Ed: <https://www.insidehighered.com/digital-learning/article/2017/05/17/where-trends-are-going-lms-market>
- Schatz, J. (2016, 10 20). *Why we Chose Vue.js*. Retrieved from GitLab:
<https://about.gitlab.com/2016/10/20/why-we-chose-vue/>
- Simpson, A. (2017, 06 4). *The True Cost Of A Learning Management System*. Retrieved from eLearning Industry: <https://elearningindustry.com/true-cost-of-a-learning-management-system>
- Wu, T.-T. (2017). *Emerging Technologies for Education*. Rome: First International Symposium SETE.

APPENDICES

APPENDIX A: USERS' MANUAL

EduForum User Manual

Table of Contents

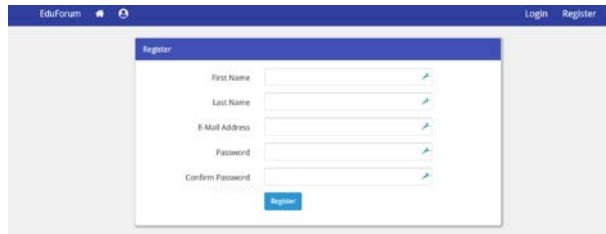
General System Information	34
Platform Administration	38
Institutional Administration	41
Course Administration	44
Student Use	46

Note : The capabilities available to users varies greatly based upon the user's position and system permissions.

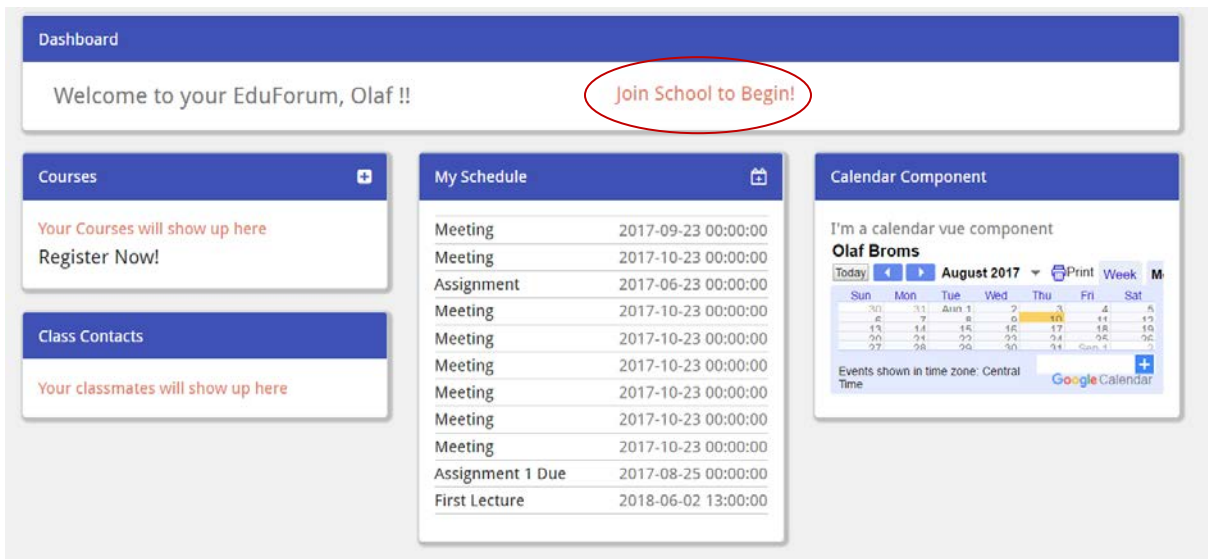
USER MANUAL 1

GENERAL SYSTEM INFORMATION

1. To begin using Eduforum, visit <http://eduforum.online/> and register for a new account



2. Once registered all users will be taken to the EduForum home page where they may conduct a number of actions but most are dependent on the user joining an institution.



Dashboard

Welcome to your EduForum, Olaf !! Join School to Begin!

Courses

Your Courses will show up here
Register Now!

Class Contacts

Your classmates will show up here

My Schedule

Meeting	2017-09-23 00:00:00
Meeting	2017-10-23 00:00:00
Assignment	2017-06-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Assignment 1 Due	2017-08-25 00:00:00
First Lecture	2018-06-02 13:00:00

Calendar Component

I'm a calendar vue component

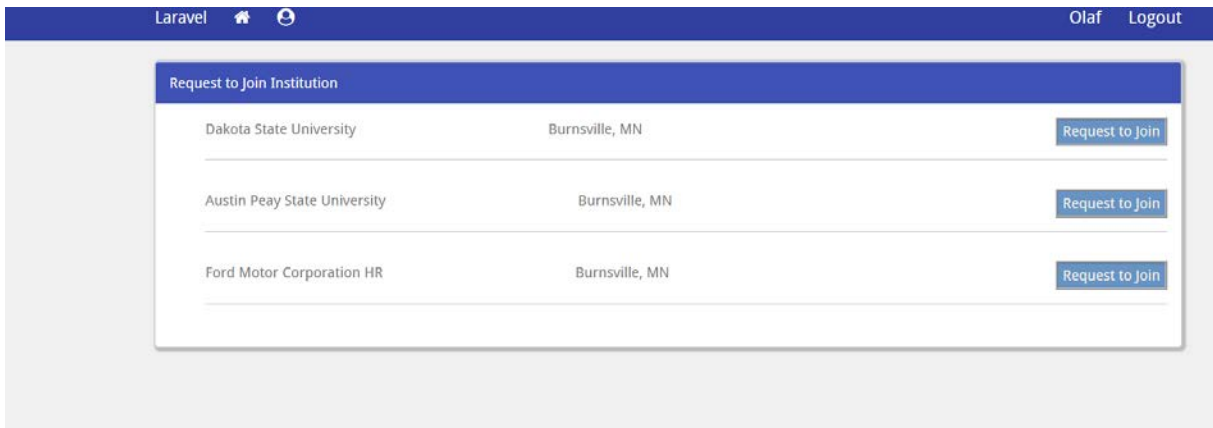
Olaf Broms

Today ← → **August 2017** Print Week M

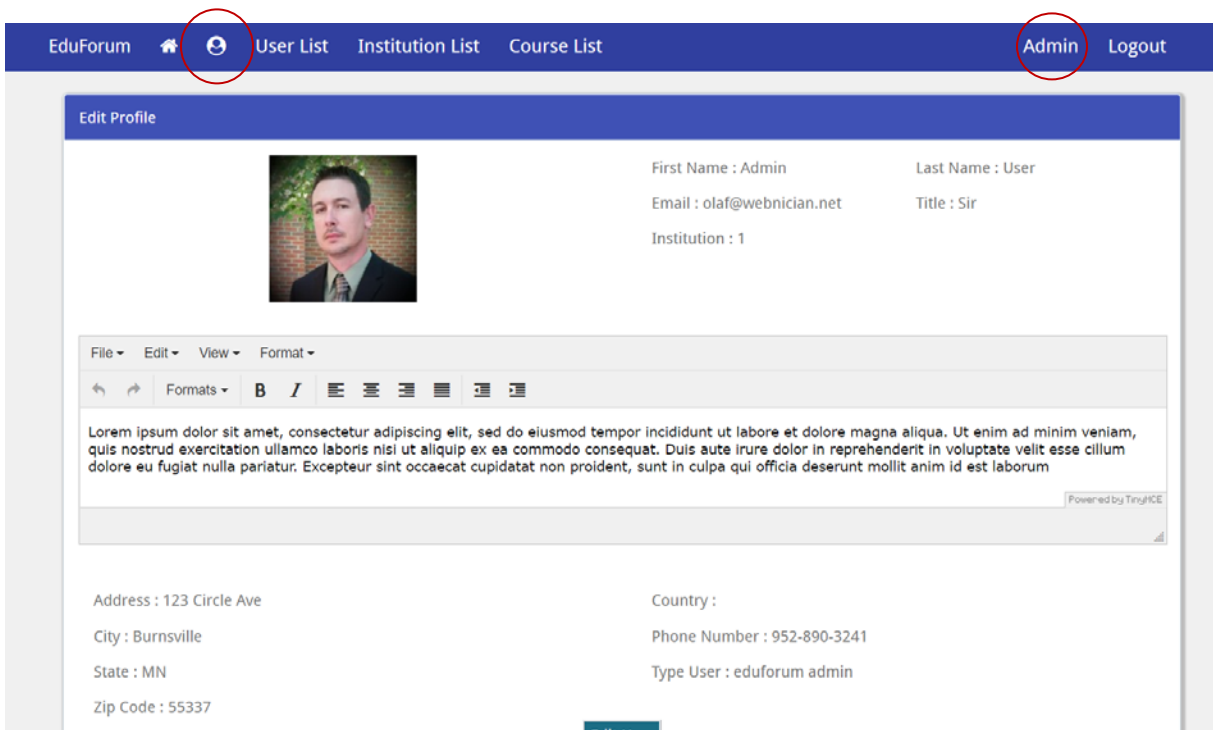
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Events shown in time zone: Central Time Google Calendar

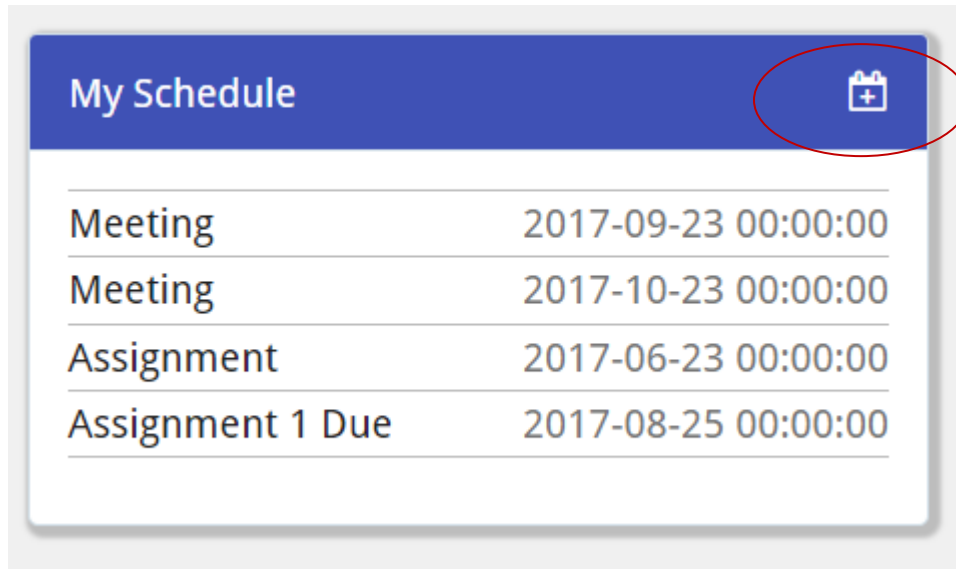
3. From here a user should select the institution that they wish to apply to



4. Once accepted to an institution a user will be able to register for courses and contact their instructors or classmates.
5. To manage a users profile they should select either their name or the Person icon in the administration bar. This will then allow the user to update various fields in their profile



6. To manage schedule items a user should select an existing schedule item to perform create, read, update, or delete operations on it. If adding a new schedule item, the Plus(+) icon should be selected.



The screenshot shows a 'Schedule Item' dialog box with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- An empty text input field at the top.
- Scheduled Time** section:
 - Start :** Input field with placeholder 'mm/dd/yyyy --:-- --'
 - End :** Input field with placeholder 'mm/dd/yyyy --:-- --'
- Description** section: Input field
- Location :** Input field with placeholder 'Location'
- Add Item** button


7. If the user is to be an institutional admin for a new institution, they should contact EduForum to have their institution added to the institution list
8. If the user is to be an institutional admin, personnel admin, course admin, or instructor at an existing institution they should contact the main institutional administrator to assign them the proper role.

USER MANUAL 2

PLATFORM ADMINISTRATION

Note : platform administrators are those that have admin privileges for the eduforum system rather than for a particular institution. There is some institution level functionality that is inaccessible to them

1. As a platform administrator, all system users are able to be edited through the User List option in the administration bar. Once the user to be edited is found in the list select them to edit or delete their profile. If a new user is desired, click the create new user button, fill in the appropriate fields, and save the new users profile.



The screenshot displays the 'User List Component' interface. At the top, there is a blue header bar with the text 'User List Component'. Below the header, the word 'Users' is displayed on the left, and a 'Create New User' button is on the right. A list of 12 users is shown, each with a number and a name, followed by a horizontal line representing a separator. The users listed are:

- 1 - Admin User
- 2 - L5rDV8W38h AL4PBONRti
- 3 - VeCw1BwM2T Ft9b95LYZF
- 4 - fjKiKjwBtY OS4WJcmUiu
- 5 - SudEiD85XK naVKfSCeRw
- 6 - 5C7UhrO6nP gPHaIT2UTq
- 7 - R1R6vaErHF A0DJ7KUzjU
- 8 - KaSb2Npj6d uNNjnVUUNZ
- 9 - YGbx6ysDPE nX2ltytptW
- 10 - M3HuFb9Ktn vL03OzUP6w
- 11 - OXjdrvrKF6 cG841qiith
- 12 - Olaf Broms

Laravel **User List** Institution List Course List Admin Logout

Edit Profile

First Name : L5rDV8W38h Last Name : AL4PBQNRti
Email : F5fWsOzvSB@gmail.com Title : bIndtUZwPD
Institution : 20

File Edit View Format
Forms B I [text alignment icons] [list icons]

Powered by TinyMCE

Address : 123 Circle Ave City : Burnsville State : MN Zip Code : 55337
Country : Phone Number : 952-890-3241 Type User : institution admin

Edit User

2. To perform create, read, update, or delete operations on institutions, select the Institution List option in the administration bar

EduForum User List **Institution List** Course List Admin Logout

Institution Component



Create New Institution

Institutions

- 1 - [Dakota State University](#)
- 2 - [Austin Peay State University](#)
- 3 - [Ford Motor Corporation HR](#)

3. To perform create, read, update, or delete operations on courses, select the Course List option in the administration bar



EduForum   User List Institution List Course List Admin Logout

Course List Component

Courses [Create New Course](#)

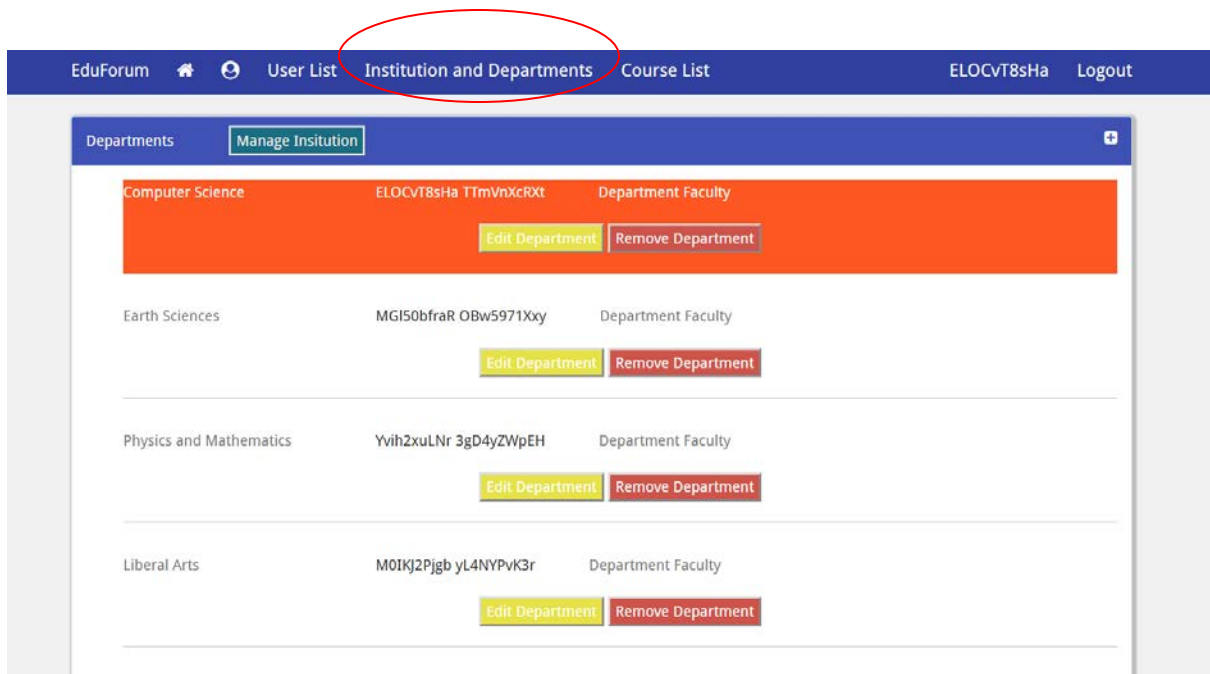
- [1 - Introduction to Programming](#)
- [2 - Data Modeling for Novices](#)
- [3 - Database Concepts and Design](#)
- [4 - Logical Relationships in Modern Web Applications](#)
- [5 - Data In Use](#)

USER MANUAL 3

INSITUTIONAL ADMINISTRATION

Note : Institutional Administrators have the capability of managing their own particular institution at a more granular level than what is available to platform administrators

1. To manage the institution select the Institution and Departments option in the administration bar



The screenshot shows the 'Institution and Departments' administration interface. The top navigation bar includes 'EduForum', 'User List', 'Institution and Departments' (circled in red), and 'Course List'. The main content area is titled 'Departments' and features a 'Manage Insitution' button. Below this, a list of departments is displayed, each with its name, a unique ID, and the role 'Department Faculty'. Each department entry includes 'Edit Department' and 'Remove Department' buttons.

Department Name	ID	Role	Actions
Computer Science	ELOCVT8sHa TTmVnXcRxt	Department Faculty	Edit Department, Remove Department
Earth Sciences	MGI50bfrar OBw5971Xxy	Department Faculty	Edit Department, Remove Department
Physics and Mathematics	Yvih2xuLnr 3gD4yZWpEH	Department Faculty	Edit Department, Remove Department
Liberal Arts	M0IKj2Pjgb yL4NYPvK3r	Department Faculty	Edit Department, Remove Department

2. To manage the institution's information, select the button labeled as Manage Institution. Update desired information and submit
3. To add a new department select the Plus(+) icon. Fill in the appropriate information and submit to add the new department to the list

EduForum User List Institution and Departments Course List ELOCVT8sHa Logout

Departments Manage Insitution

Computer Science	ELOCVT8sHa TmVnXcRxt	Department Faculty	Edit Department	Remove Department
Earth Sciences	MGI50bfrAR OBw5971Xxy	Department Faculty	Edit Department	Remove Department
Physics and Mathematics	Yvih2xuLnr 3gD4yZWpEH	Department Faculty	Edit Department	Remove Department
Liberal Arts	M0IKj2Pjgb yL4NYPvK3r	Department Faculty	Edit Department	Remove Department

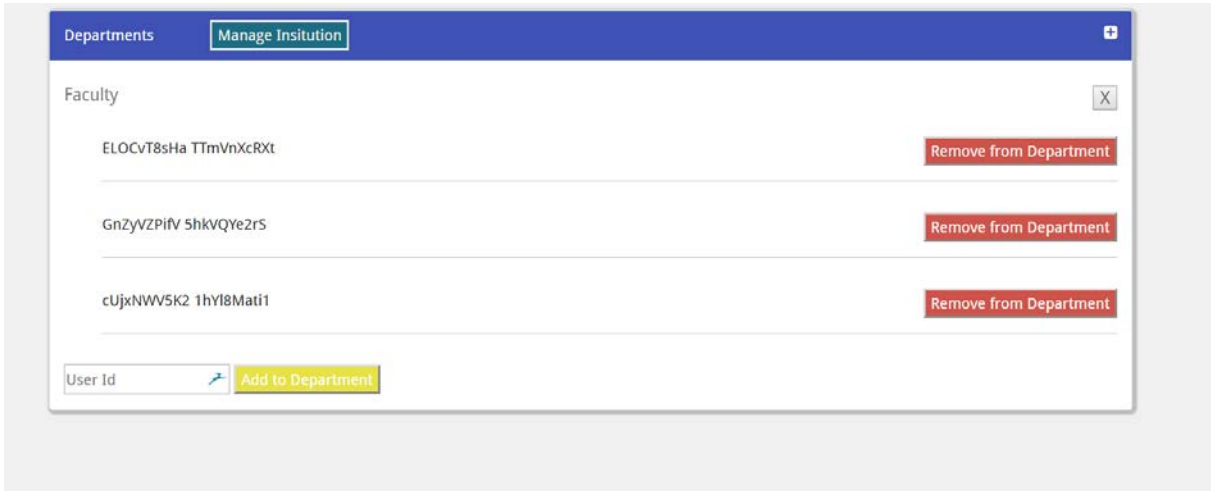
4. To edit a department select the Edit Department button, update the desired information and then save the department. To remove a department, select the Remove Department button

EduForum User List Institution and Departments Course List ELOCVT8sHa Logout

Departments Manage Insitution

Computer Science	ELOCVT8sHa TmVnXcRxt	Department Faculty	Edit Department	Remove Department
Earth Sciences	MGI50bfrAR OBw5971Xxy	Department Faculty	Edit Department	Remove Department
Physics and Mathematics	Yvih2xuLnr 3gD4yZWpEH	Department Faculty	Edit Department	Remove Department
Liberal Arts	M0IKj2Pjgb yL4NYPvK3r	Department Faculty	Edit Department	Remove Department

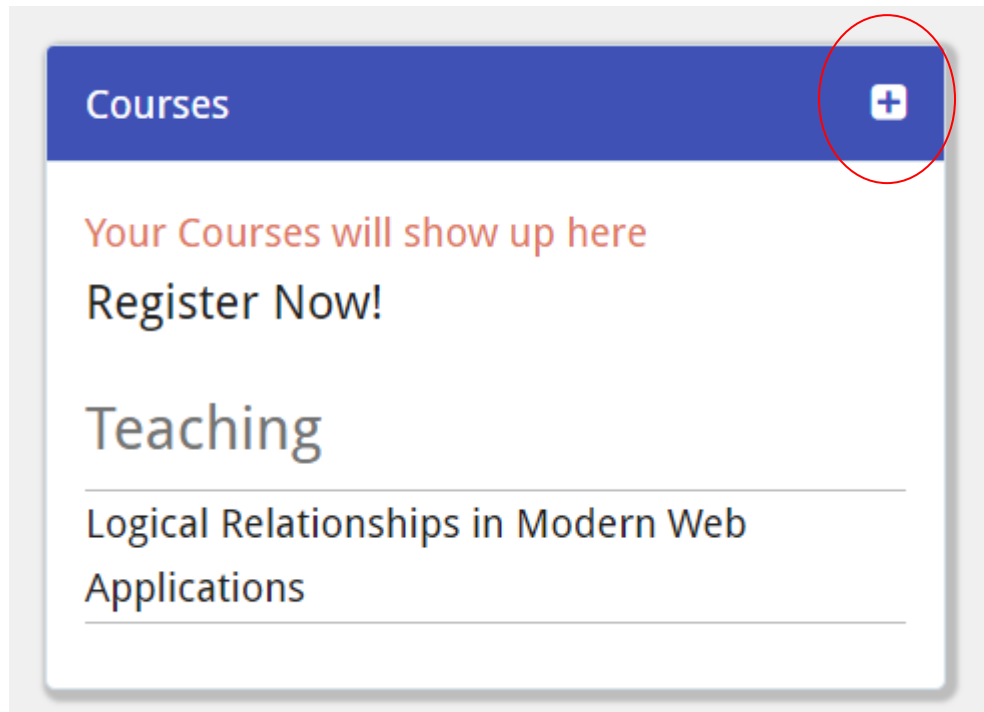
5. To manage department faculty, select the department faculty link for the appropriate department. This will open the department faculty screen where users can be added and removed from the department



USER MANUAL 4

COURSE ADMINISTRATION

1. To access a course, the appropriate course should be selected in the Courses menu



2. Selecting the course will then take the user to the corresponding class page

The screenshot shows a dashboard titled "Data In Use" with a "Save Layout" button. It contains five widgets, each with a red number in a box pointing to it:

- 1**: "Class Documents" widget showing a list with "File 0".
- 2**: "File Upload" widget with a "Name File" input, "Choose File" button, "Choose Files" button (with "No file chosen" text), "Upload File to Course" button, and "Upload" button.
- 3**: "Class List" widget showing "Instructor" information: "cUjxNwV5K2" and "billy".
- 4**: "Class Schedule" widget showing "First Lecture" on "2018-06-02 13:00:00".
- 5**: "Assignment List" widget showing two assignments: "Assignment 1" (2017-08-10 14:00:00) and "Assignment 2" (2017-08-20 13:00:00).

3. Item 1 in the above figure lists documents added to the course by an instructor or a course administrator and allows for students to download them. Item 2 allows for instructors or course administrators to upload new documents to the course. Item 3 lists the course instructor as well as the registered students. Item 4 allows instructors or course administrators to perform create, read, update, or delete operations on course specific schedule items. Students also have the ability to view these. Item 5 allows for instructors or course administrators to perform create, read, update, or delete operations on course assignments. Students may also view these.

USER MANUAL 5

STUDENT USE

Note : students have the additional capability to submit assignments in addition to being able to update their profile and register for courses.

1. To submit an assignment, the user should access the course that they are registered for in the Courses Menu

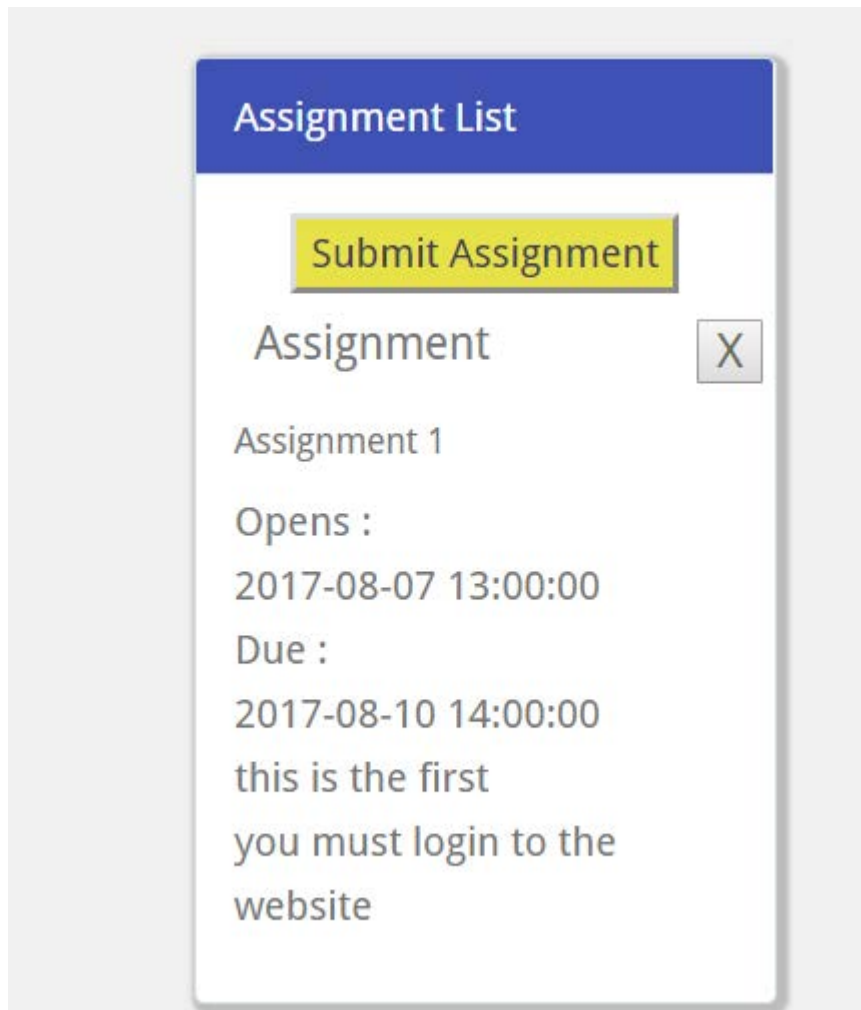
The screenshot displays the EduForum dashboard for a user named Billy. The dashboard is divided into several sections:

- Dashboard Header:** Welcome to your EduForum, billy !! | Dakota State University | ONLINE LEARNING CONSORTIUM
- Courses Menu:** Registrations (highlighted with a red circle), Data In Use, Class Contacts, Data In Use, Instructor : cUjxNwV5K2 1hY18Mat1, Contact
- My Schedule:** A list of events including Meetings, Assignments, and a First Lecture with their respective dates and times.
- Calendar Component:** A calendar for August 2017 showing the current date and a list of events.

Event	Date/Time
Meeting	2017-09-23 00:00:00
Meeting	2017-10-23 00:00:00
Assignment	2017-06-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Meeting	2017-10-23 00:00:00
Assignment 1 Due	2017-08-25 00:00:00
First Lecture	2018-06-02 13:00:00

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	Sen. 1	2

2. Once in the course, the user should select the appropriate assignment and click the submit assignment button



3. The user should then add the submission content or upload a file in the submission form and select the Submit Assignment button

Submit Assignment

Assignment Title :	Assignment open date :
this is the first	2017-08-07 13:00:00
Assignment :	Assignment due date :
you must login to the website	2017-08-10 14:00:00

Assignment Submission :

File ▾ Edit ▾ View ▾ Format ▾

↶ ↷ Formats ▾ **B** *I* [List Icons]

Powered by TinyMCE

[Submit Assignment](#) [Back](#)

APPENDIX B: SYSTEM TECHNICAL DOCUMENTATION

Project Gantt Chart

The project Gantt chart was used to organize sprint tasks and to ensure that tasks were being completed in a sensible manner

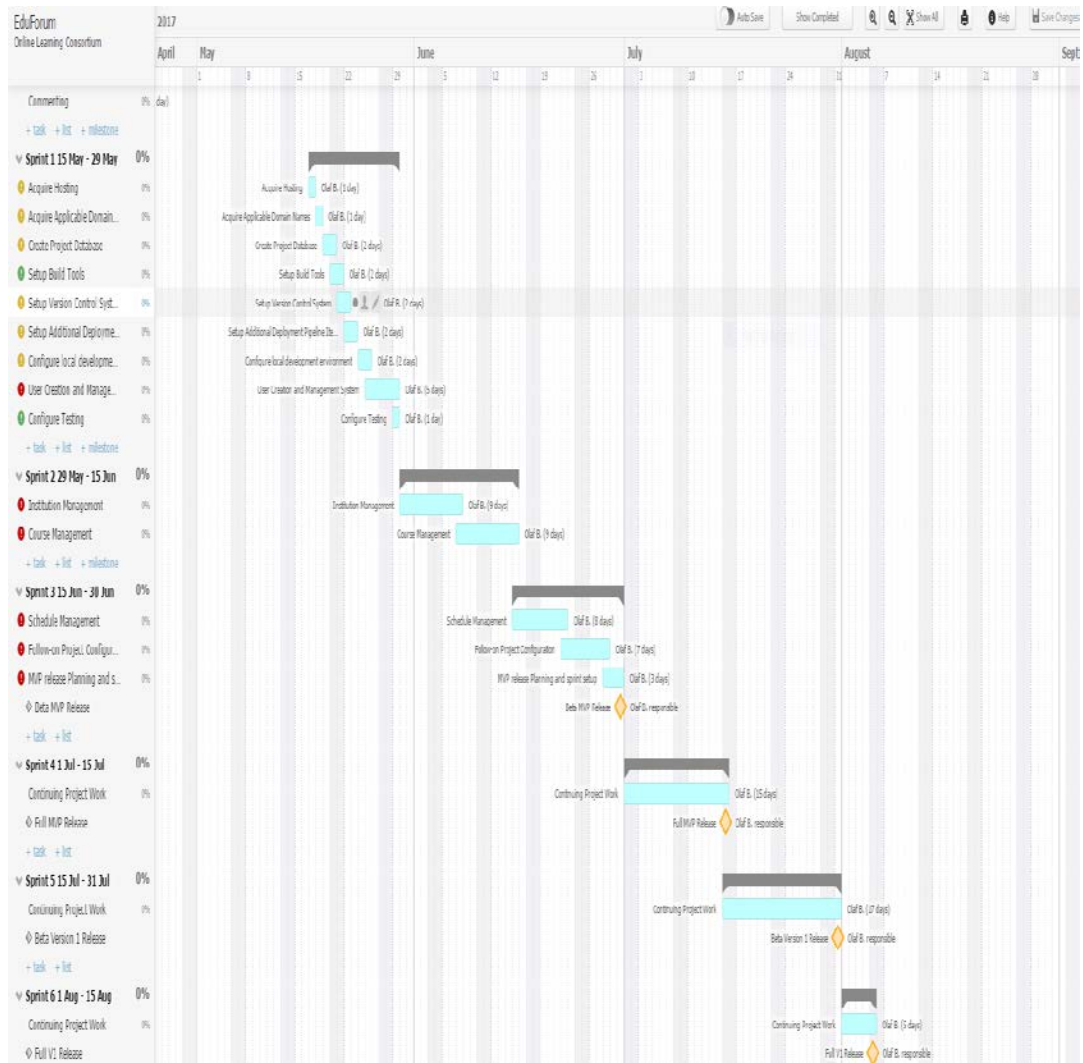


Figure 2- Project Gantt Chart

Project Work Breakdown Structure

The work breakdown structure was used to ensure that project work was being completed in a streamlined manner.

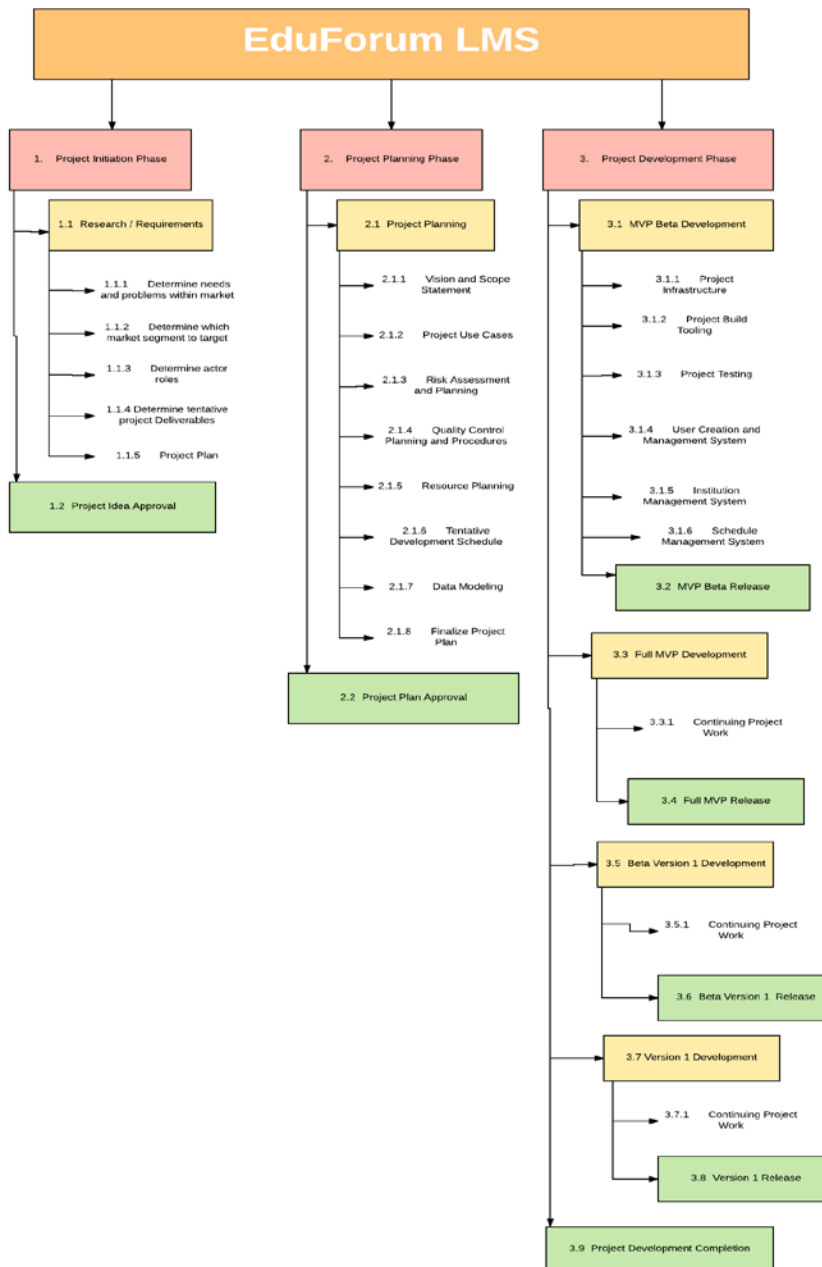


Figure 3- Work Breakdown Chart

Project Entity-Relationship Diagram (database model)

The entity relationship diagram was used to create a tentative database layout and to plan the flow of data between the different entities within the data model.

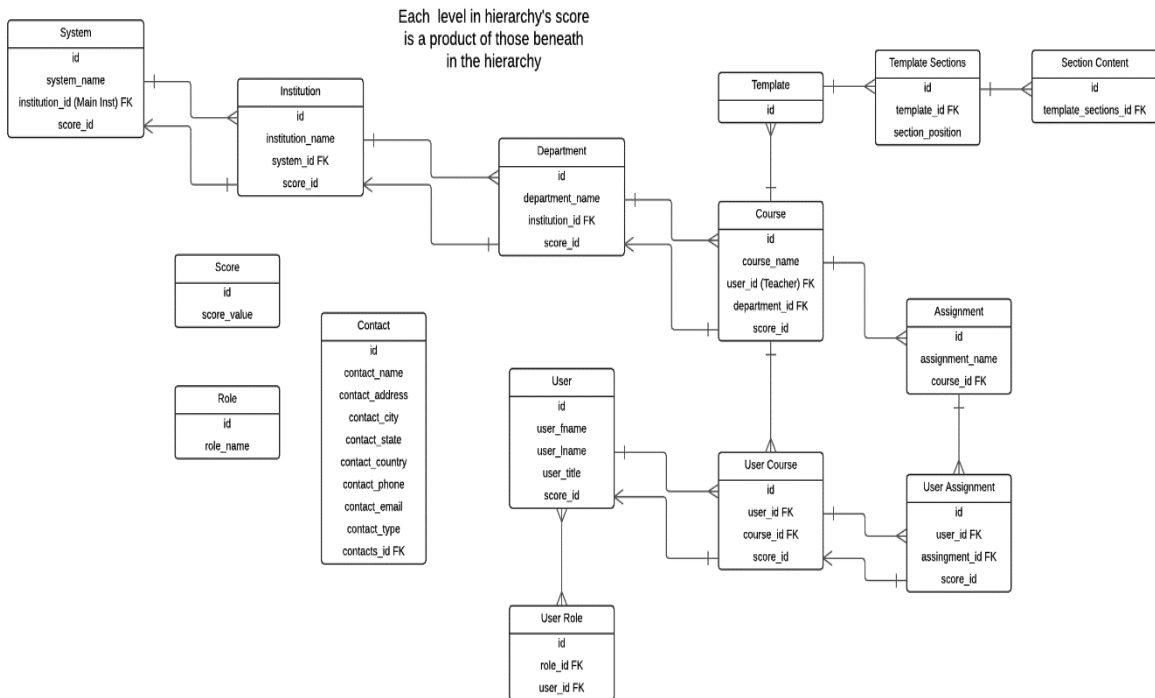


Figure 4- Entity Relationship Diagram

Database Tables

This diagram is a snapshot of the version 1 database table structure

Name ^	Rows	Size	Created	Updated	Engine	Comment	Type
assignments	3	48.0 KiB	2017-07-20 21:27:32		InnoDB		Table
contacts	0	16.0 KiB	2017-07-20 21:27:31		InnoDB		Table
courses	4	48.0 KiB	2017-07-20 21:27:31		InnoDB		Table
course_join_requests	3	48.0 KiB	2017-07-20 21:27:31		InnoDB		Table
institutions	3	48.0 KiB	2017-07-20 21:27:31		InnoDB		Table
institution_contacts	0	16.0 KiB	2017-07-05 22:38:05		InnoDB		Table
inst_contacts	3	32.0 KiB	2017-07-20 21:27:31		InnoDB		Table
inst_departments	4	48.0 KiB	2017-07-20 21:27:31		InnoDB		Table
inst_join_requests	2	48.0 KiB	2017-07-20 21:27:31		InnoDB		Table
migrations	24	16.0 KiB	2017-05-16 02:16:59		InnoDB		Table
password_resets	0	16.0 KiB	2017-07-20 21:27:31		InnoDB		Table
permissions	24	64.0 KiB	2017-07-20 21:27:31		InnoDB		Table
permission_role	60	48.0 KiB	2017-07-20 21:27:31		InnoDB		Table
permission_user	0	16.0 KiB	2017-07-20 21:27:31		InnoDB		Table
roles	10	48.0 KiB	2017-07-20 21:27:31		InnoDB		Table
role_user	2	48.0 KiB	2017-07-20 21:27:31		InnoDB		Table
schedule_items	10	64.0 KiB	2017-07-20 21:27:31		InnoDB		Table
users	11	32.0 KiB	2017-07-20 21:27:31		InnoDB		Table
user_assignments	0	16.0 KiB	2017-07-20 21:27:32		InnoDB		Table
user_contacts	11	32.0 KiB	2017-07-20 21:27:31		InnoDB		Table
user_courses	6	48.0 KiB	2017-07-20 21:27:31		InnoDB		Table
user_depts	2	48.0 KiB	2017-07-20 21:27:32		InnoDB		Table
user_insts	5	48.0 KiB	2017-07-20 21:27:31		InnoDB		Table
user_preferences	5	32.0 KiB	2017-07-20 21:27:31		InnoDB		Table

Figure 5- Project Database Design

Example User Interface Code(Vue.JS)

```

<div class="container">
  <div class="row">
    <div class="panel panel-default" >
      <div class="panel-heading" v-model="theClass"><h3>{{ theClass.course_name }}</h3></div>
    </div>
  </div>

  <form action="/classpreferences/update" method="POST">
    <input type="hidden" name="_token" :value="csrf">
    <input type="hidden" v-for="(block, index)"
    |in theuiblock" :name="'block'+index" v-model="block.title" />
    <button class="btn-info" type="submit">Save Layout</button>
  </form>

  <div class="row theuserinterface full-screen-user-interface" >
    <draggable v-model="theuiblock" @start="drag=true" @end="drag=false">
      <div v-for="(block, index)"
      in theuiblock" :id="'position'+index" v-bind:class="getClasses(index)" style="...">
        <component v-bind:is="block.title" v-bind="theProps(block)" >
          <!-- component changes when vm.currentView changes! -->
        </component>
      </div>
    </draggable>
  </div>
  <div class="row theuserinterface mobile-user-interface" >
    <div v-model="theuiblock" >
      <div v-for="(block, index)"
      in theuiblock" :id="'position'+index" v-bind:class="getClasses(index)" style="...">
        <component v-bind:is="block.title" v-bind="theProps(block)" >
          <!-- component changes when vm.currentView changes! -->
        </component>
      </div>
    </div>
  </div>
</div>

```

```

<div class="container">
  <div class="row">
    <div class="col-lg-12 col-md-12 col-sm-12 ">
      <div class="panel panel-default">
        <div class="panel-heading">Class List</div>

        <div class="panel-body">
          <div v-model="instructor">Instructor<br/>{{ instructor.fname }}</div>
          <ul>
            <li v-for="student in studentlist">{{ student.student.fname }}</li>
          </ul>
        </div>
      </div>
    </div>
  </div>
</div>

```

Example Controller Code (Laravel)

```
public function StudentAssignment()
{
    if(Auth::check()) {
        $user = Auth::user();
        $input = Input::get();

        $assignment = self::get_assignment($input['assignment_id']);
        $course      = Course::get_course($input['course_id']);

        return view('view: /assignments.student-assignment')
            ->with('thecourse', $course)
            ->with('theuser', $user)
            ->with('assign', $assignment);
    }
    else{
        return redirect( to: 'home');
    }
}

public function StudentSubmissions()
{
    if(Auth::check()) {
        $user = Auth::user();
        $input = Input::get();

        dd($input);

        $course      = Course::get_course($input['course_id']);
        $course      = \GuzzleHttp\json_encode($course);

        $the_assign = $this->get_assignment($input['assignment_id']);

        $assignments = UserAssignment::get_user_assignments_by_assignment_id($input['assignment_id']);
        $iterator = 0;
        foreach ($assignments as $assignment)
        {
            $user = User::get_user_by_id($assignment['user_id']);
            $assignments[$iterator]['user'] = $user;
            $iterator ++;
        }
        $assignments = \GuzzleHttp\json_encode($assignments);

        return view('view: /assignments.student-assignment-list')
            ->with('thecourse', $course)
            ->with('theuser', $user)
            ->with('assignments', $assignments)
            ->with('the_assign', $the_assign)
            ;
    }
    else{
        return redirect( to: 'home');
    }
}
```


Example Model Code (Laravel)

```
public static function update_assignment($input, $content)
{
    $assignment = self::get_assignment_by_id($input['assignment_id']);
    $assignment->course_id = $input['course_id'];
    $assignment->assignment_name = $input['assignname'];
    $assignment->assignment_description = $input['assigndesc'];
    $assignment->assignment_open_date = $input['open_date'];
    $assignment->assignment_due_date = $input['end_date'];
    $assignment->assignment_content = $content;
    $assignment->save();
}
```

```
public static function get_join_requests_by_inst_id($id)
{
    $joinrequests = InstJoinRequests::where('institution_id', $id)->get();

    $iterator = 0;
    foreach($joinrequests as $request)
    {
        $joinrequests[$iterator]['theuser'] = User::get_user_by_id($request['user_id']);
        $iterator++;
    }
    return $joinrequests;
}
```

Automatic Deployments Bash Scripting (CodeShip)

Webnician/EduForum

Tests
Deploy
Environmen

CUSTOM SCRIPT

```
#!/bin/bash
# Deploy to via pushing to a remote git repository.
#
# Add the following environment variables to your project configuration and make
# sure the public SSH key from your projects General settings page is allowed to
# push to the remote repository as well.
#
# Include in your builds via
# \curl -sSL https://raw.githubusercontent.com/codeship/scripts/master/deployments/git_push.sh | bash -s
#REMOTE_REPOSITORY=${REMOTE_REPOSITORY:?'You need to configure the REMOTE_REPOSITORY environment variable!'}
#REMOTE_BRANCH=${REMOTE_BRANCH:?'You need to configure the REMOTE_BRANCH environment variable!'}
set -e
① git fetch --unshallow || true
# All necessary Commands
git config --global user.name "Codeship Deploy"
git config --global user.email "olaf@webnician.net"
#git remote add deploys "${REMOTE_REPOSITORY}"
git remote add deploys git@107.170.53.95:/var/repo/site3.git
git add .
git commit -m "Deploy"
git push --force deploys master
#git push --force deploys "${CI_COMMIT_ID}":master
#git push "${REMOTE_REPOSITORY}" "${CI_COMMIT_ID}:${REMOTE_BRANCH}"
#ssh deploy@107.170.53.95
#git remote add deploying "${REMOTE_REPOSITORY}"
#git push deploying
```

Automatic Virtual Machine creation and Testing (CodeShip)

I want to create my own custom commands

Setup Commands

```
phpenv local 7.0
mkdir -p ./bootstrap/cache
composer install --prefer-source --no-interaction
cp -v .env.example .env
php artisan key:generate
php artisan config:cache
```

These are the commands that should be run before the tests are executed. Use and mock

Configure Test Pipelines (1 / 1 Used)

[+ Add Pipeline](#)

Test Commands

```
vendor/bin/phpunit
```

Version Control System Repository (Github)

Webnician / EduForum Unwatch 2 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

eduforum shall be an open source learning management system Edit

Add topics

161 commits 1 branch 0 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

blaster32blaster user assignment submission Latest commit 070d2f1 9 days ago

.idea	UI add packages add/remove entrust roles	3 months ago
app	user assignment submission	9 days ago
bootstrap	rebase proj	4 months ago
config	add assignments and further styling	20 days ago
database	add assignments and further styling	20 days ago
public	user assignment submission	9 days ago
resources	user assignment submission	9 days ago
routes	user assignment submission	9 days ago
storage	rebase proj	4 months ago
tests	testing? and add tinymce api key	20 days ago
.env.example	rebase proj	4 months ago
.env.travis	update to check tests	3 months ago
.gitattributes	rebase proj	4 months ago
.gitignore	rebase proj	4 months ago
.travis.yml	update to check tests	3 months ago
LICENSE	rebase proj	4 months ago

Project Management Platform (Teamwork PM)

+ ▾ Inbox 28 tasks - [Expand](#)

- ▾ Sprint 5 15 Jul - 31 Jul


Milestone: [Beta Version 1 Release](#) (a month ago)

No tasks in this list yet.

+ Add a Task 12 Completed

- ▾ Sprint 6 1 Aug - 15 Aug

Milestone: [Full V1 Release](#) (25 days ago)

 **Olaf B.** Continuing Project Work (Started Wed Jul 12th → Due Sun Jul 16th) [Infrastructure](#) **x**

+ Add a Task

+ ▾ QA / Acceptance Testing 0 tasks - [Expand](#)

+ ▾ Bottleneck / Stalled 0 tasks - [Expand](#)

+ ▾ Bugs 9 tasks - [Expand](#)